

RealServer[®]
Administration
and
Content Creation Guide
Version 4.0

Progressive Networks, Inc.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Progressive Networks, Inc.

© 1995-1997 Progressive Networks, Inc. All rights reserved.

RealAudio and RealAudio logo are registered trademarks of Progressive Networks, Inc.

RealVideo and the Real logo are trademarks of Progressive Networks, Inc.

PowerPC are trademarks of IBM Corporation.

Adobe Premiere is a trademark of Adobe Systems, Inc.

Apple and Macintosh are trademarks of Apple Computer, Inc.

DolbyNet is a trademark of Dolby Laboratories, Inc.

Dolby Digital AC-3 audio system manufactured under license from Dolby Laboratories Licensing Corporation.

Macromedia is a registered trademark and Shockwave is a trademark of Macromedia, Inc.

Netscape Netsite is a trademark of Netscape Communications Corp.

O'Reilly Website NT is a trademark of O'Reilly Associates, Inc.

Sonic Foundry and Sound Forge are trademarks of Sonic Foundry, Inc.

SunOS and Solaris are trademarks of Sun Microsystems, Inc.

UNIX is a registered trade mark of X/Open Company Limited.

Windows and Windows NT are trademarks of Microsoft Corp.

SE100 is a trademark of Creative Labs

Wakeboard Multimedia Pro is a trademark of Digital Video Arts

CD30 is a trademark of Miro

Bravado 1000 is a trademark of Truevision

Osprey 1000 is a trademark of Multimedia Access

Seagate AV drives is a trademark of Seagate

Sipro is a trademark of Sipro Lab Telecom, Inc.

Progressive Networks, Inc.
1111 Third Avenue, Suite 2900
Seattle, WA 98101 USA
www.real.com

Contents

Section 1: Introduction

Introduction	7
Using this Guide	8
Overview	13
New Features	13
RealVideo and RealAudio Clips	14
Components of a RealServer System	15
RealVideo and RealAudio Files and Metafiles	16
Delivering a RealVideo or RealAudio Clip	17
Delivering Synchronized Multimedia	18
Image Maps	20
Title, Author, and Copyright	20
Updating RealVideo and RealAudio Clips	21
Live Encoding and Live Broadcasting	21
System Requirements	22
RealServer Requirements	22
RealVideo Encoder Requirements	25
RealAudio Encoder Requirements	28
Installation on Windows Platforms	31
Installing RealPlayer	31
Installing RealServer	32
Testing RealServer	33
Using RealServer as a Service	35
RealServer Directories and Files	37
Upgrading RealServer from a Previous Version	39
Installing the System Manager	41
Installing RealVideo Encoder	43
Installing RealAudio Encoder	45
Installation on UNIX Platforms	48
Installing RealPlayer	48
Installing RealServer	49
Testing RealServer	55
RealServer Directories and Files	57

Upgrading RealServer from a Previous Version	61
Starting and Testing the System Manager	63
Installing RealAudio Encoder	64

Section 2: Administration

Configuration File	67
Editing the Configuration File.....	70
Configuring Basic Server Settings	77
General Settings.....	78
Configuring RealServer for Smart Networking.....	89
Configuring Access and Error Logging	91
Configuring E-mail and Usage Thresholds	95
Limiting Server Bandwidth and the Number of Simultaneous Connections	102
Security.....	104
System Manager Settings	107
Configuring Web Servers to Work with RealServer	109
Configuring Advanced Server Settings	115
Splitting	115
Configuring Remote Licensing	125
Configuring for ISP Hosting.....	129
Configuring for Intranets.....	134
Problem Solving	136
Using the Access and Error Log Files	136
Access and Error Log Messages	138
RealServer Access Log	138
RealServer Error Log	144
Troubleshooting RealServer	151

Section 3: Server Operation

Server Operation.....	153
Starting RealServer	153
Stopping RealServer	158
Monitoring Performance	160
Command Reference.....	164

Section 4: RealVideo Content Creation

Encoding RealVideo Clips.....	179
System Requirements.....	179
Source Files	179

Encoding Templates.....	185
RealVideo Encoder Plug-in for Adobe Premiere	188
Using RealVideo Encoder for Windows and Macintosh	191
Creating Templates	198
Editing Video Files.....	202
Editing Input Files	202
Editing RealVideo Files in Windows	202
Command Line Editing of RealVideo Files	210
Image Maps	218
Modifying RealVideo File Descriptions	222
Bandwidth Negotiation	223
RealVideo Content Creation for Internet Appliances and Set-top Boxes.....	227

Section 5: RealAudio Content Creation

Encoding RealAudio Clips	229
System Requirements.....	229
Source Files	230
Choosing an Encoding Algorithm	232
Encoding RealAudio files with Sound Forge	234
Using RealAudio Encoder for Windows.....	244
Using the Encoder for Macintosh	250
Using RealAudio Encoder for UNIX.....	258
RealAudio Batch Encoding	261
Editing Audio Files.....	262
Editing Input Files	262
Editing RealAudio Files	262
Modifying RealAudio File Descriptions	268
Bandwidth Negotiation	271

Section 6: Live Broadcasting

Live Broadcasting.....	278
Advertising Your Live Event	279
Delivering Live Content	281
Setting Up RealServer.....	281
EncoderPassword	282
EncoderTimeout	283
RealVideo Live Broadcasts	283
RealAudio Live Broadcasts.....	295
Archiving Live Broadcasts	304

LiveFilePassword.....	306
LiveFileSize.....	306
LiveFileTarget.....	307
LiveFileTime.....	308
URL.....	308
BandwidthEncoding.....	309
LiveFileBandwidthNegotiation.....	310
Simulating a Live Broadcast.....	311
Simulated Live Broadcast.....	311
InputFile.....	312
OutputFile.....	312
ServerHost.....	313
ServerPassword.....	313
ServerPort.....	314

Section 7: Multicasting

Multicasting.....	315
Overview.....	315
Configuring RealPlayer for Multicast Reception.....	318
Configuring RealServer for Multicast Broadcasting.....	319

Section 8: Configuring Your Web Site

Configuring Your Web Site.....	326
Making the Most of Your Content.....	326
Metafiles.....	327
HTTP Streaming.....	331
Custom Controls for RealAudio and RealVideo.....	332
Synchronized Multimedia.....	349
Creating a RealVideo Events File.....	350
Creating a RealAudio Events File.....	352
Using Synchronized Multimedia with Bandwidth Negotiation.....	355
Frames: The Basics.....	355
Index.....	359

Introduction

RealServer[®] is today's most powerful solution for streaming real-time live and on-demand video and audio over the Internet and corporate intranets. With the award-winning RealVideo and RealAudio applications, RealServer provides a proven, reliable platform for multimedia streaming that allows you to scale up and grow as your network and use expands.

Real-time delivery means that you do not have to wait while a file downloads; the clip plays as it is delivered. You have complete control over the clip; you can pause, move forward and back, and start or stop at any time.

RealVideo formats are optimized for low- to medium-speed connections including 14.4 and 28.8 Kbps modems, ISDN, T1 and LAN. You can also view RealVideo files stored locally on your computer.

RealAudio formats are optimized for low- to medium-speed connections including 14.4 and 28.8 Kbps modems and ISDN. You can also listen to RealAudio files stored on your local computer.

This guide explains how to configure and operate RealServer and how to create and manage content for RealVideo and RealAudio systems. The content includes all of the clips and multimedia presentations delivered by RealVideo and RealAudio systems, and the support files that make it all work smoothly.

Using this Guide

RealServer Administration and Content Creation Guide is intended for IS managers, server administrators, Web masters and others providing Web pages for the Internet and intranet.

Organization

This manual is organized in eight sections as follows:

Section 1: Introduction

Explains system requirements and installation of RealServer, RealAudio Encoder, and RealVideo Encoder.

Overview: Provides an overview of RealServer including new features, and a description of the components of a RealServer system.

System Requirements: Explains the system requirements for RealServer, RealAudio Encoder, and RealVideo Encoder on Windows 95, and Windows NT, and UNIX platforms.

Installation Instructions: Describes installation procedures for RealPlayer, RealServer, RealAudio Encoder, and RealVideo Encoder on Windows 95, Windows NT, and UNIX platforms.

Section 2: Administration

Explains how to administer RealServer.

Editing the Configuration File: Introduces the server configuration file and describes how to edit the file using a text editor or the System Manager.

Configuring Basic Server Settings: Describes the settings in the server configuration file that control the basic operation of RealServer.

Configuring Web Servers to Work with RealServer: Describes how to set the correct MIME type to make the user's web browser play the contents of a media file with RealAudio or RealPlayer.

Configuring Advanced Server Settings: Describes the settings in the server configuration file that control more advanced server features. These features are not included in the basic RealServer License.

Problem Solving: Provides troubleshooting information including using the access and error log files.

Section 3: Server Operation

Describes how to start and stop RealServer and how to monitor performance.

Starting RealServer: Describes how to start RealServer both manually and automatically.

Stopping RealServer: Describes how to stop RealServer.

Monitoring Performance: Describes how you can monitor a Server running on any platform.

Command Reference: Provides reference information for commands used with RealServer.

Section 4: RealVideo Content Creation

Explains how to create and manage video and multimedia content for delivery by RealServer and a Web server. The chapters are:

Encoding RealVideo Files: Information about how to prepare files for encoding, which encoding algorithm to use and how to encode a static file using RealVideo Encoder or Adobe Premiere.

Editing RealVideo Files: Explains how to use the editing tools to change the static information in a file, alter the data, and examine the contents of a file.

Section 5: RealAudio Content Creation

Explains how to create and manage audio content for delivery by RealServer and a Web server.

Encoding RealAudio files: Information about how to prepare files for encoding, which encoding algorithm to use and how to encode a static file using RealAudio Encoder or Sound Forge.

Editing Audio Files: Explains how to edit encoded files using the Racut, Rapaste and Rax utility programs.

Bandwidth Negotiation: Detailed instructions and examples of how to use bandwidth negotiation

Section 6: Live Broadcasting

Live Broadcasting: Provides an overview of live broadcasting, information on how to advertise your live event.

Delivering Live Content: Explains how to set up RealServer to deliver live content, and how to connect to the encoder.

RealVideo Live Broadcasts: Explains how to deliver live video, including setting up RealServer and advertising your event on Timecast: RealMedia Guide.

RealAudio Live Broadcasts: Explains how to deliver live audio, including setting up the RealServer and advertising your event on Timecast: the RealAudio Guide.

Archiving Live Broadcasts: Describes how to save a live broadcast on the service for re-broadcast later.

Simulating a Live Broadcast: Describes how to play a recorded media file as if it were being broadcast live.

Section 7: Multicasting

Provides a brief overview of multicasting and describes how to configure RealServer and RealPlayer for multicasting.

Configuring RealPlayer for Multicast Reception: Describes how to configure RealPlayer so clients can receive a multicast broadcast.

Configuring RealServer for Multicast Broadcasting: Describes how to configure RealServer so clients can receive a multicast broadcast.

Section 8: Configuring Your Web Site

Explains the many ways you can deliver RealVideo and RealAudio from your Web site; including metafiles, HTTP streaming, ActiveX object and JavaScript.

Making the Most of Your Content: Describes how you can get the most out of your RealAudio and RealVideo content, and educate your Web site's visitors about RealAudio and RealVideo.

Metafiles: Explains how to create metafiles which contain information needed to establish a connection between your RealServer and your listener's RealPlayer and to initiate playback.

HTTP Streaming: Explains how to stream RealAudio and RealVideo clips from a World Wide Web server to provide short RealAudio and RealVideo content to a limited number of users.

Custom Controls for RealVideo and RealAudio: Describes how you can place individual interactive components, such as a play button or image window, anywhere on your page.

Synchronized Multimedia: Discusses how to create a synchronized multimedia presentation.

Conventions

This manual uses the following conventions:

Command	Monospace font represents commands to be typed or information displayed on the screen.
<file name>	Angle brackets show where to insert information, such as the name and location of a file.
[]	Square brackets show optional command arguments.
Bold	Bold font is used for names of files, directories, commands, and options.

Overview

Superb audio and video quality, high transmission reliability, great broadcast scalability, and flexible modes of delivery have made the RealServer the market leader for professional live and on-demand streaming media on the Internet or your intranet. RealServer 4.0 provides on award-winning RealVideo and RealAudio technology and offers many exciting new features.

New Features

- **Video**
RealServer 4.0 support video streaming .
- **Smart Networking**
RealServer 4.0 provides support for Smart Networking. This feature allows a server to send content via uni-cast or multi-cast and use either UDP, TCP, or HTTP depending on current network conditions and the presence of firewalls or firewall proxies. Content is now available to all users and will be sent using the most efficient means possible.
- **Multiple IP Support**
RealServer 4.0 provides support for binding to multiple IP addresses. This allows the server to be configured to transmit on multiple IP addresses and/or use multiple network interfaces. It can also be used to provide support for Smart Networking on machines running Web servers.
- **UDP-Based Splitting**
RealServer 4.0 now supports a new more robust splitting mechanism. This allows a live stream to be broadcast to multiple servers over the Internet. This version of splitting differs from the 3.0 implementation by using UDP and being driven by the server configuration rather than player request.
- **Stream Thinning**
This feature allows RealServer to dynamically adjust the video data rate to match the client's current throughput. This allows continuous, un-

interrupted playback over bad connections. RealServer automatically decreases the amount of data being streamed to the Player. This feature insures the highest quality video at a given bandwidth.

- **New and Improved Server Control Application and Installer**
RealServer 4.0 now features a new improved installer and a graphical server control application (NT only).
- **RMTools**
RealServer 4.0 includes RMTools. RMTools allows you to edit, cut and paste RealVideo and RealAudio files. It also allows you to edit the title, author, copyright fields, and gives you the ability to change audio tracks.

RealVideo and RealAudio Clips

A RealVideo clip is a file or live broadcast containing sound and video encoded in RealVideo formats. A RealAudio clip is a file or live broadcast containing sound encoded in one of the RealAudio formats. These formats are highly compressed to deliver the best possible sound and video over a limited-bandwidth connection.

Because there is no single best format for delivering audio and video, RealServer system provides several formats that are optimized differently for different kinds of content. You can choose to provide a clip in one or more formats based on the type of content and the available bandwidth. For example, you would use a different format to deliver speech over a 14.4 Kbps modem than you would to deliver a music video over an ISDN connection.

Components of a RealServer System

RealServer System is a client-server system. The server provides the content to the client over a network. The basic components of a RealServer system are:

RealPlayer - The client program that enables users to see and listen to RealVideo and RealAudio clips. Special purpose player components enable ActiveX and Netscape Plug-in capabilities.

RealVideo Encoder - The program that creates RealVideo clips. The input to this program can be a digitized audio clip, a live audio signal, a digitized video clip or a live video signal.

RealAudio Encoder - The program that creates RealAudio clips. The input to this program can be a digitized audio file or a live audio signal.

RealVideo Tools - The tools programs that can be used to modify RealVideo and RealAudio clips, including RMTTools, RMMerge, RMEEdit, RMPaste, RAX, and RACut.

RealServer - The server program that delivers RealVideo and RealAudio clips over a network. One RealServer can deliver clips to many Players at the same time.

Web Browser - The client program that enables users to find most RealVideo and RealAudio clips. The Web browser is also used to display the visual part of Synchronized Multimedia presentations. RealVideo system works with all popular Web browsers.

Web Server - The server program that delivers Web pages. Typically, RealVideo and RealAudio clips are accessed by clicking a link on a Web page. The Web server also delivers the visual part of Synchronized Multimedia presentations. HTTP streaming of RealVideo and RealAudio files enables content providers to stream RealVideo and RealAudio clips from a World Wide Web server. While this method is not as robust, it provides a reasonable method for providing short RealVideo and RealAudio content to a limited number of users. RealServer System works with all popular Web servers.

RealVideo and RealAudio Files and Metafiles

RealServer System uses several file types, each identified by a specific file extension. The files and their file extensions are:

RealVideo clip (.rm) - Audio and video encoded to RealVideo format. This file is created with RealVideo Encoder and delivered by RealServer. The .rm files can contain multiple streams, including audio, video, image maps, and events.

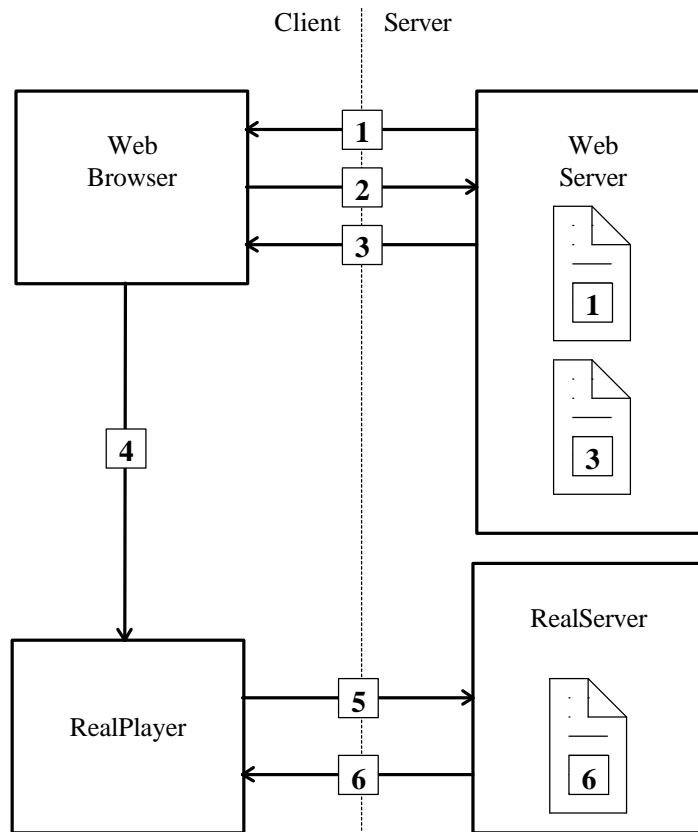
RealVideo or RealAudio metafile (.ram) - The file that connects a Web page to one or more RealVideo or RealAudio clips. The metafile is located on a Web server and is linked by a Web page. The metafile contains the URL of one or more clips located on a RealServer.

RealPlayer Plug-in metafile (.rpm) - The same as a metafile, but used with RealPlayer Plug-in for Netscape Navigator and Internet Explorer 3.0 and later.

RealAudio clip (.ra) - Audio encoded to RealAudio format. This file is created with RealAudio Encoders and works with RealPlayer, RealAudio Player, and RealServer.

Delivering a RealVideo or RealAudio Clip

The following figure shows the components of RealServer system used to deliver a typical RealVideo or RealAudio clip. The numbers in the figure match the numbered steps following the figure.



RealServer Delivery System

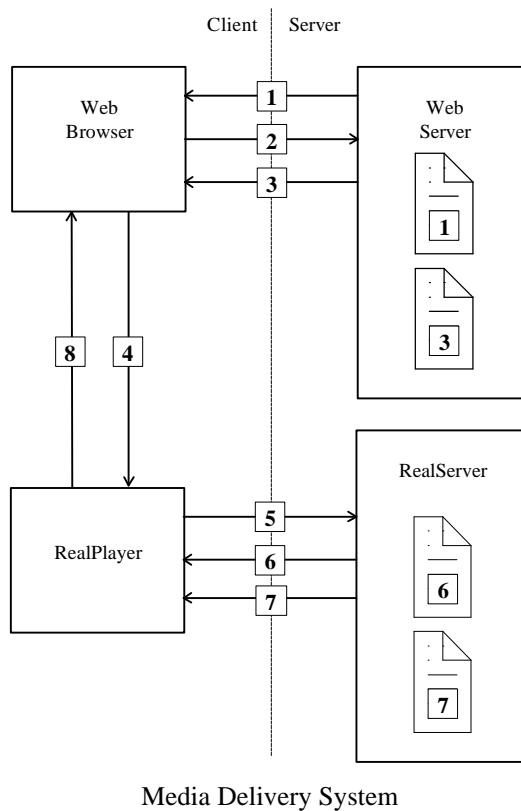
1. The Web browser displays a Web page that contains a link to a metafile.
2. The user clicks the link. The Web browser requests the metafile from the Web Server.
3. The Web server delivers a metafile to the Web browser. Based on the .ram file extension, the Web server sets the MIME type of the file to **audio/x-pn-RealAudio**. For files with a .rpm file extension (RealPlayer Plug-in), the Web server sets the MIME type of the file to **video/x-pn-RealVideo-plugin** or **audio/x-pn-RealAudio-plugin**.
4. The Web browser looks up the MIME type of metafile. Based on the MIME type, the Web browser starts RealPlayer as a helper application and passes it the metafile.
5. RealPlayer reads the first URL from the metafile and requests it from RealServer.
6. RealServer begins streaming the requested RealVideo or RealAudio clip to RealPlayer.

Note No messages pass between RealServer and the Web server. The Web browser provides the URL of RealVideo or RealAudio clip to RealPlayer. RealPlayer does not require a Web browser to function; users can enter the URL of a .ra, .rm or .ram file directly into RealPlayer, or use the Preset or Scan buttons on RealPlayer Plus.

Delivering Synchronized Multimedia

Synchronized Multimedia allows the creation of multimedia presentations. Synchronized Multimedia delivery works like regular RealAudio or RealVideo delivery, with one extra step between RealPlayer and the Web browser.

The following figure shows the components of RealServer System used to deliver a typical Synchronized Multimedia presentation. The numbers in the figure match the numbered steps following the figure.



1. The Web browser displays a Web page that contains a link to a metafile.
2. The user clicks on the link. The Web browser requests the metafile from the Web Server.
3. The Web server delivers a metafile to the Web browser. For files with a .ram file extension, the Web server sets the MIME type of the file to **audio/x-pn-RealAudio**. For files with a .rpm file extension (RealPlayer Plug-in), the Web server sets the MIME type of the file to **RealVideo-plugin** or **audio/x-pn-RealAudio-plugin**.
4. The Web browser looks up the MIME type of metafile. Based on the MIME type, the Web browser starts RealPlayer as a helper application and passes it the metafile.

5. RealPlayer reads the first URL from the metafile and requests it from RealServer.
6. RealServer begins streaming the requested RealAudio or RealVideo clip to RealPlayer.
7. RealServer concurrently notifies RealPlayer when an event should occur. For more information see “Using Synchronized Multimedia with Bandwidth Negotiation” on page 355.
8. RealPlayer sends the URL for the event to the default Web browser for viewing.

Image Maps

RealVideo allows users to interact with video content using "click-able" image maps. Image maps are fully customizable; actions can be connected to rectangular, circular or polygonal content regions and can be varied over specific user defined time intervals.

Title, Author, and Copyright

RealAudio and RealVideo clips include text strings for the title, author, and copyright. This text is displayed by RealPlayer when the clip is played. Although the player usually labels the text as title, author, and copyright, the player displays whatever text you choose to supply.

Enter the title, author, and copyright text when you encode a clip. You can override the values encoded with the clip using the metafile. You can also change the text using RMEdit and RMTTools.

For information about changing the Title, Author or Copyright text strings in encoded files, refer to “Modifying RealVideo File Descriptions” on page 222.

Updating RealVideo and RealAudio Clips

When you encode a RealVideo or RealAudio clip, the original file is not modified. RealVideo Encoder creates a new file with a .rm file extension. RealAudio Encoder creates a new file with a .ra file extension. It is important to note that encoding a RealVideo or RealAudio clip is a one-way process; you cannot convert a RealVideo or RealAudio file back into the original source format. If you want to be able to encode to other formats in the future, you need to archive the original source.

It is possible to modify various attributes of RealVideo and RealAudio clips without re-encoding them. For information about changing the Title, Author or Copyright text strings in encoded files; or about enabling or disabling PerfectPlay and Selective Record, refer to “Modifying RealVideo File Descriptions” on page 222.

Live Encoding and Live Broadcasting

The source of a RealVideo or RealAudio clip can be prerecorded or live input. The encoded RealVideo or RealAudio clip can be stored as a file for later use, broadcast live over a network or it can be simultaneously stored as a file and broadcast live. The last option allows you to archive the live broadcast.

All versions of RealVideo Encoder can encode a live input source. Live broadcasting requires a license and RealServer.

System Requirements

This chapter describes the hardware and software requirements for RealServer, RealAudio Encoder, and RealVideo Encoder.

RealServer Requirements

To run RealServer, you need:

- A computer running one of the supported operating systems
- Space on the computer's hard disk for the RealServer and the media files you plan to offer
- A network connection of sufficient bandwidth to serve your users
- A Web server that supports configurable MIME types
- Sufficient memory and processor capacity

Memory and CPU Usage

RealServer requires approximately 6 MB of available RAM plus 40 KB RAM for each simultaneous stream. To support 100 simultaneous connections requires approximately 10 MB of available memory.

RealServer has a modest CPU impact. A 100-stream RealServer operating on a 120 MHz Pentium computer consumes less than 30% of the CPU cycles. With enough network bandwidth, the same computer can deliver at least 400 28.8 streams simultaneously.

Disk Space Requirements

The RealServer program files require about 2 MB of disk space. You also need disk space for the content files you are serving. See “System Requirements” on page 24 for video encoding, and on page 28 for audio encoding.

Bandwidth Requirements

Bandwidth requirements for audio and video depend on the encoding/decoding formats used. For video, see “Bandwidth Negotiation” on page 223. For audio, see “Bandwidth Negotiation” on page 271.

Compatible Web Servers

Typically, users access media files using links embedded in World Wide Web pages. Therefore, you need to have a Web server installed and configured to recognize certain MIME types. The details of this configuration are discussed in the chapter entitled “Configuring Web Servers to Work with RealServer” on page 109. Although you need a Web server to make the best use of RealAudio and RealVideo, you do not need to install it on the same machine as your RealServer.

RealServer can be configured to work with any Web server that supports configurable MIME types (which most Web servers support) RealServer has been tested with the following Web servers:

- CERN HTTPD (v 3.0)
- EMWAC HTTPS 0.96
- HTTPD4Mac
- Mac HTTP
- Microsoft Internet Information Server (IIS)
- NCSA HTTPD (v 1.3, 1.4, 1.5)
- Netscape Netsite and Netscape Enterprise Server
- O'Reilly Website NT

- Webstar and Webstar PS
- Spinner 1.0b12 - 1.0b15 / Roxen 1.0
- Apache 1.1.1

To view a current list of World Wide Web servers tested for compatibility with RealServer, see the Progressive Networks Web site at:

<http://www.real.com>

RealVideo Encoder Requirements

RealVideo Encoder is available for the following platforms:

- Windows 95 and Windows NT
- Macintosh

Windows 95 and Windows NT Requirements

RealVideo Encoder is designed to work in the Windows 95 and Windows NT (3.51 or later) environments. The following table explains which hardware is required for static encoding. For information about additional requirements for delivering live content, refer to “Live Encoding” on page 26.

Requirement	Minimum	Recommended
CPU: Windows 95/NT	486/66 DX	Pentium 120
RAM	16 MB	32 MB
Hard Disk space (software)	4 MB	4 MB
Hard Disk space (data)	500 MB	1 GB
Color Display	16-bit	24-bit (TrueColor)
Video Capture Card (not required to encode an AVI or Quick Time file)	Any card that is compatible with Video for Windows.	
Sound Card	16-bit sound card	

Special Hard Drives for Capturing Video

Capturing video puts a burden on your computer's ability to write to your hard drive. Seagate manufactures special hard drives for capturing video called AV drives. If you do not use one of these special drives, frames may be dropped during the capture process when the hard drive re-calibrates itself periodically. If you experience dropped frames with a standard drive, you can re-capture the video.

Live Encoding

The quality of live encoding depends on the power of your computer system. Encoding at higher quality takes longer and requires greater computing resources than encoding at a lower quality. Slower machines give reasonable results in Slide Show mode, at 1 fps or less. The following lists the requirement for encoding at low and high frame rates.

Requirement	Low Frame Rate (<2 fps)	High Frame Rate (.>2 fps)
Computer	Pentium 166 MHz	Dual Pentium 200 MHz
OS	Windows 95	Windows NT
RAM	32 MB	64 MB
Capture Card	Osprey 100, Osprey 1000, ISVR III	Osprey 100, Osprey 1000
Audio Card	Any quality audio card	Any quality audio card
Hard Drive	1 GB	1 GB

The following lists the recommended settings for a variety of different content types:

Content Type	Total Bit rate	Audio codec	Frames/sec	Quality Setting
General	20 Kbps	6.5 Kbps	2	100
General	20 Kbps	6.5 Kbps	7.5	70
Newscast	20 Kbps	6.5 Kbps	4	100
Newscast	45 Kbps	8.5 Kbps	10	1
Music	20 Kbps	12 Kbps	5	100
Music	45 Kbps	16 Kbps	5	70

For Low frame-rate live encoding, less than 3 fps, a P166 or better computer, running Windows 95, is required. Any capture board which captures a Video for Windows format of YUV9 or RGB24 (preferred) will work. If you are not getting the desired frame rate, lower the quality setting.

Macintosh Requirements

RealVideo Encoder for the Macintosh has the following requirements:

Macintosh Requirement	Minimum	Recommended
CPU	Any PowerPC	604 PowerPC CPU
RAM	16 MB	32MB
Operating System	System 7.1.2	System 7.5.5

68K Macintosh computers are not supported.

RealAudio Encoder Requirements

RealAudio Encoder is available for the following platforms:

- Windows 95 and Windows NT
- UNIX (Sun Solaris 2.5, SGI Irix 5.3, Linux, FreeBSD, SunOS 4.1.4)
- Macintosh

Windows 95 and Windows NT Requirements

The following table explains which hardware is required for specific Encoder uses on Windows 95 and Windows NT platforms. Note that different hardware is required for encoding from a file and for encoding a live audio stream.

Requirement	File Encoding	Live Broadcasting
CPU	486/66 DX	Pentium/586, 75 MHz
RAM	8 MB	8 MB required 16 MB recommended
Hard Disk Space (software)	1 MB	1 MB
Sound Card	Depends on the encoding algorithm chosen. The following table shows the preferred sampling rate for each encoding algorithm.	

Encoding Algorithm	Sampling Rate
RealAudio 2.0 - 14.4	8 kHz
RealAudio 2.0 - 28.8	8 kHz
RealAudio 3.0 - 28.8 Mono, full response	11.25 kHz
RealAudio 3.0 - 28.8 Mono, medium response	11.25 kHz
RealAudio 3.0 - 28.8 Mono, narrow response	8 kHz
RealAudio 3.0 - 28.8 Stereo	8 kHz
RealAudio 3.0 - ISDN Mono	22.05 kHz
RealAudio 3.0 - ISDN Stereo	16 kHz
RealAudio 3.0 - Dual ISDN Mono	44.1 kHz
RealAudio 3.0 - Dual ISDN Stereo	32 kHz

UNIX Requirements

The RealAudio Encoder for UNIX is designed to work with several UNIX operating systems:

Operating System	Version
Solaris	2.5
SunOS	4.1.4
Linux	1.2.8
Linux	1.3.x
Irix	5.3
FreeBSD	2.1.5

Macintosh Requirements

The minimum system requirements for the Macintosh RealAudio Encoder are as follows:

- Apple System 7.1 or later
- 68030 Macintosh with a floating point co-processor (FPU)

For best results, recommended systems are:

- PowerMac
- Quadra 700 or faster

Installation on Windows Platforms

Setup programs make installing RealServer, the System Manager, the RealAudio Encoder, and the RealVideo Encoder, quick and easy. This chapter explains the entire installation and upgrade process for Windows NT platforms.

If you are upgrading a previous version of the server, read the upgrade procedure described in “Upgrading RealServer from a Previous Version” on page 39 before you begin the installation procedure.

Installing RealPlayer

You will need a Player to test the operation of the server. You can install RealPlayer on the same computer on which you are going to install the server or on any Windows, UNIX, or Macintosh computer that is networked to the Server. The computer on which you install RealPlayer must have a sound card.

To install RealPlayer:

1. With your Web browser, go to the Progressive Networks Web site:
<http://www.real.com>
2. Follow the instructions for downloading RealPlayer
3. Run the setup program that you have downloaded The setup program installs the RealPlayer software and automatically plays a test clip on RealPlayer. RealPlayer is correctly installed if it plays the test clip.

Installing RealServer

You can install RealServer on an Intel-based Windows NT computer or a DEC Alpha-based Windows NT computer.

RealServer is available directly from the Progressive Networks site on the World Wide Web:

<http://www.real.com>

After you fill out the download form, Progressive Networks sends you an e-mail message that contains a customer name and a license key needed to install the server. You will be asked for the license information during installation. We recommend that you cut and paste the customer name and license key from the e-mail. This will ensure the information is entered exactly.

Installing RealServer is a simple process of following the setup RealServer screens. When you have completed the minimum information required for installation, the **Finish** button will become active. Clicking **Finish** will use the default selections for the remaining screens. If you have any question about the information on a screen, Help text is available by clicking the **Help** button.

To install RealServer:

1. With your Web browser, go to the URL in the confirmation e-mail you received.
2. Follow the instructions for downloading RealServer.
3. Run the RealServer setup program that you have downloaded. The RealServer setup screen displays.
4. Follow the screen prompts to complete the installation.

RealServer is also available on CD-ROM, which is mailed to you, along with this manual, after you place an order for the server software. If you purchased RealServer from a reseller, locate the license key label on the box. You will be asked for the license information during installation. Take care to enter the license information exactly.

If you are installing RealServer from the CD-ROM:

1. Insert the RealServer CD-ROM into your hard disk.
2. Double-click the Install RealServer icon to install the program files on your hard disk.
3. Double-click the **RealServer** icon. The RealServer setup screen displays.
4. Follow the screen prompts to complete the installation.

Testing RealServer

After you install RealServer you should test it by starting the Server and playing a sample media file.

Starting RealServer on Windows NT

To start RealServer from RealServer Control Center:

1. Double-click on the RealServer Control Center icon on your desktop, or from Start, select **Programs**.
2. Follow the instructions in “Starting RealServer on a Windows NT Computer” on page 33.

To start RealServer manually from the command line:

1. Change to the directory into which you installed RealServer (**pnserver** by default).
2. Start RealServer by entering:

```
start bin\pnserver server.cfg
```
3. RealServer does not return any messages to indicate that it has started.

Playing a Sample Clip

You can test RealServer operation by playing a clip in the **content** directory:

1. Start RealPlayer on any Macintosh, Windows or UNIX computer that can access your RealServer through a network.
2. From the RealPlayer File menu, click **Open Location**.
3. In the **URL** text, enter the path of the media file as:

```
pnm: // <my.pnserver> : <port> /welcome.ra
```

where:

<my.pnserver> is the DNS name or the IP address of the computer with RealServer installed.

<port> is the port number to use for testing.

welcome.ra is a sample audio file automatically installed into your RealServer's **content** directory.

If RealPlayer plays the media file, then RealServer is installed correctly.

If the media file does not play at all, or if the performance or playback quality is poor, see the Troubleshooting chapter for information to help you diagnose and correct the problem. Also check your log files for clues. To learn about log files, see “Using the Access and Error Log Files” on page 136.

Note By default, media files are located in the **Real\Server** directory. In general, you should not put media files in directories below your Web Server. The RealServer is independent of and does not communicate with the Web server.

Stopping RealServer

Once RealServer is running correctly, you need to stop it before changing configuration settings as explained in the next chapter.

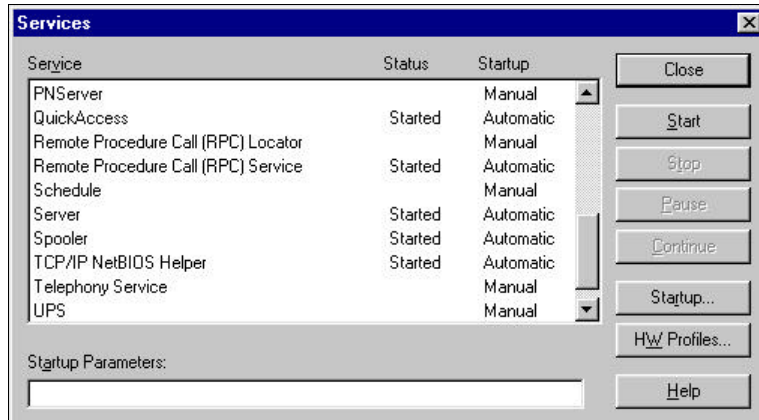
To stop RealServer for Windows NT from the command line, press CTRL+C in the window used to start the server. If you are running RealServer from the Server Control Center, from Server, select **Stop**.

Using RealServer as a Service

If you install RealServer as a Windows NT service, you can set RealServer to automatically start when the computer starts. You can also use the Services dialog box to manually start and stop RealServer.

To set RealServer to Autostart:

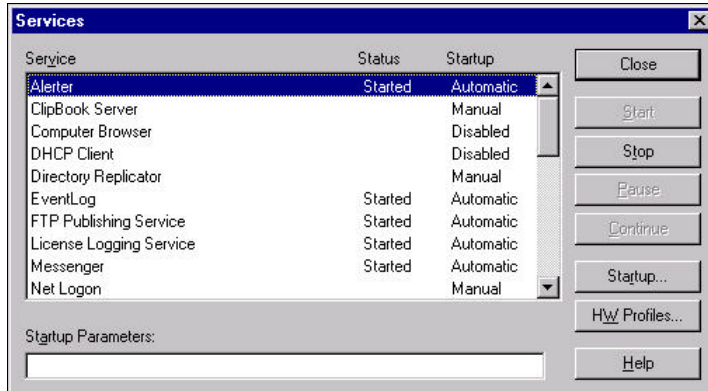
1. Open the Windows NT Control Panel.
2. Double-click the Services icon.



3. In the Service list, double-click **Pnserver**.
4. Under Startup Type, select **Automatic**.

To start RealServer from the Services dialog box:

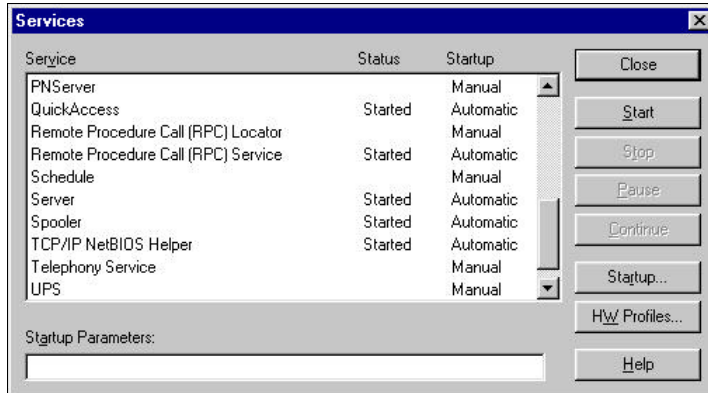
1. Open the Windows NT Control Panel.
2. Double-click the Services icon.



3. In the Service list, select Pnserver.
4. Click the **Start** button.

To stop RealServer from the services dialog box:

1. Open the Windows NT Control Panel.
2. Double-click the Services icon.



3. In the Service list, select Pnserver.
4. Click the **Stop** button.

RealServer Directories and Files

Real\Server is the default directory for the server files. The **bin**, **log**, and **content** are subdirectories below the default server directory.

Real\Server Directory

The **Real\Server** directory contains the following ASCII text files:

File	Description
server.cfg	The server configuration file, which contains the configuration settings for the server.
license.txt	Your license agreement.
readme.txt	Information about server installation and operation.
problem.txt	Information about known issues in RealServer.

Bin Directory

The **bin** directory contains the following executable files:

File	Description
crtsvc	The create service utility, which you can use to install RealServer as a Windows NT service.
delsvc	The delete service utility, which you can use to uninstall RealServer as a Windows NT service.
pnservice	The RealServer executable.
raconv	The bandwidth negotiation file converter, which converts RealAudio files to a bandwidth negotiation names scheme.
rmfile	The live file creation utility, which creates (archives) files from live broadcasts.
svrctrl.exe	Server Control Application, which you can use to start and stop RealServer.

Log Directory

The **log** directory contains the following log files, which are in ASCII text format:

File	Description
log	This file contains a message indicating that the log directory is the log directory.
Pnaccess.log	The access log, which logs information about clients that have connected to the server.
Pnerror.log	The error log, which logs informational and error messages about server operation.

Content Directory

The **content** directory stores your media files. Sample media files are installed in the **content** directory.

Upgrading RealServer from a Previous Version

If you are upgrading from a previous version of RealServer, you should install the new version next to, and not on top of, the old version. Once you have installed and tested the new RealServer, you can then replace your old version.

Installing the New Version

To install without replacing an existing RealServer:

1. Install the new version in a different directory or folder other than your existing RealServer. This prevents you from affecting your existing RealServer.
2. Note the PnaPort entry in your existing RealServer configuration file (**server.cfg** by default). If the PnaPort entry is set to 7070 or is not present, RealServer is using the default port 7070. You need to use a different port number for testing RealServer. To use a different port number, add or edit the following line to your **server.cfg** file after installing:

```
PnaPort      7072
```

Be sure to specify a port that is not used by another configuration parameter such as **ResolverPort**. If you have problems starting RealServer on this port because another application is using it, try using a different port number.

3. To send a test URL to the new RealServer, you must add :7072 to the pnm URL. This makes the test URL become:

```
pnm://<my.server>:7072/<your content>.ra
```

Note On Windows NT you cannot run two versions of RealServer as a Service. Instead, run the test copy of RealServer from a command window until you are ready to remove the previously installed server.

Configuring the New Version

After you have tested your installation of RealServer, you can then have it duplicate serving your existing content.

1. Copy the **BasePath** entry from the **server.cfg** file of your existing RealServer to the **server.cfg** file for the new version. It is best to use an absolute base path for the **BasePath** entry.
2. Send a test URL to the new Server. Remember to add :7072 to the pnm URL. This makes the test URL become:

```
pnm://<my.server>:7072/welcome.rm
```

Moving the New Version into Production

When you are satisfied the RealServer is supplying your existing content, you can stop your old server, and move the new server into production.

To move the new version into production:

1. Stop both instances of RealServer.
2. Rename the directory containing the old installation.
3. Rename the directory containing RealServer, using the original name of the old server software.
4. Merge all appropriate settings from your existing configuration file into your new configuration file.

5. Set the **PnaPort** entry back to its original value.
6. Change your Web page links back to their original port number.

Installing the System Manager

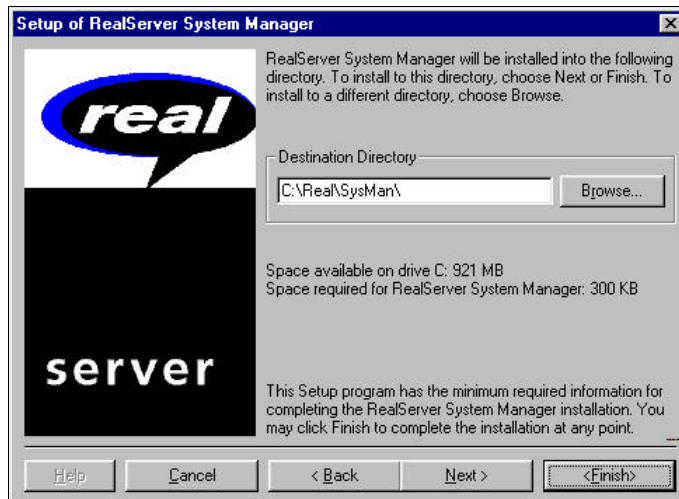
The System Manager is included in the distribution of the RealServer files. With the System Manager, you can monitor the server's performance, check and modify the server configuration, view a list of clients that are connected to the server, and perform other administrative activities. It is therefore highly recommended that you install the System Manager immediately after you install RealServer.

Note You can install System Manager on a computer other than the one on which RealServer is running, provided that you can connect System Manager to RealServer over the network. You can install System Manager on a Windows NT computer or a Windows 95 computer.

To install System Manager:

1. If necessary, start the setup program by double-clicking the setup file
2. The setup program starts.
3. Click **Install RealServer System Manager**.
4. Close any other applications that may be running. Then click the **Next** button.

5. If you want to change the default folder for the System Manager files, specify a new folder.



6. Click the **Next** button. The System Manager files are installed in the specified folder.
7. If a previous version of RealServer System Manager is detected, you will be prompted to uninstall the previous version. It is recommended that you perform the uninstallation procedure. If you skip the uninstallation, and are installing to a different directory, no files will be deleted. Click **Uninstall**.
8. Select RealServer System Manager from the list of components to be uninstalled. Click the **OK** button.

9. You will be prompted with a list of the RealServer System Manager components successfully installed on your computer. Click the **OK** button.



10. You are returned to the RealServer System Manager screen. Click the **Exit** button.

Installing RealVideo Encoder

You can download the RealVideo Encoder from the Progressive Networks Web site or install it from the RealServer CD-ROM.

To download the install program from the Internet:

1. With a Web browser, download the installer from the Progressive Networks Web site:

<http://www.real.com>

2. When prompted, click the **Save to Disk** button and specify the temporary directory for the installation program. Note where you saved it.

3. In File Manager or Explorer, double-click the icon to activate the automated installation process.

To install the RealServer setup program from CD-ROM:

1. Insert the CD-ROM into the drive.
2. In File Manager or Explorer, double-click **setup.exe** in the **\video\encoder\intel-nt** directory of the RealServer CD-ROM to activate the automated installation process.

To install the encoder:

1. Close any other applications that may be open Click the **Next** button.
2. To accept the terms and conditions for installing the Encoder, click the **Next** button.
3. Enter you name, company or organization name, and e-mail address. Then click the **Next** button.
4. Enter the directory into which you want to install the Encoder files.
5. If you are installing over an existing RealVideo Encoder for which you have defined custom styles (templates), select **Keep Existing Styles** if you want to preserve the custom styles.
6. To finish the installation, click the **Finish** button.
7. Click the **OK** button.

The following files are installed in the RealVideo Encoder directory (**\Real\RVEncode** by default):

File	Description
encllc	An ASCII text file that contains the license agreement for the encoder.
EX-RM.prm	Adobe Premiere plug-in.
graphic.gif	RealVideo logo.

File	Description
Readme.htm	Information about Encoder installation and operation.
Rmdump.exe	A utility used to examine the contents of a RealVideo file.
rmedit.exe	A utility used to modify the title, author, copyright, comment, mime type, or stream name of a RealVideo file.
rmevents.dll	A dynamic link library used by the rmmerge utility.
rmimap.dll	A dynamic link library used by the rmmerge utility.
rmmerge.exe	A utility used to create a Synchronized Multimedia presentation.
rmpaste.exe	A utility used to re-sort from bandwidth to timestamp order.
rvbatch.exe	Lets you run the encoder from a batch file as a command-line tool.
rvencode.exe	The RealVideo Encoder executable.
rvslta	Simulated Live Transfer Agent.

Installing RealAudio Encoder

You can download the RealAudio Encoder from the Progressive Networks Web site, or you can install the RealAudio Encoder from the RealServer CD-ROM.

To download the install program from the Internet:

1. With a Web browser, download the installer from the Progressive Networks Web site:

<http://www.real.com>
2. When prompted, click the **Save to Disk** button and specify the temporary directory for the installation program. Note where you saved it.
3. In File Manager or Explorer, double-click the icon to activate the automated installation process.

To install the RealServer setup program from CD-ROM:

1. Insert the CD-ROM into the drive.
2. In File Manager or Explorer, double-click **setup.exe** in the **\audio\encoder\intel-nt** directory of the RealServer CD-ROM to activate the automated installation process.

To install the Encoder:

1. Close any other applications that may be open Click the **Next** button.
2. To accept the terms and conditions for installing the Encoder, click the **Next** button.
3. Enter your name, company or organization name, and e-mail address. Then click the **Next** button.
4. Enter the directory into which you want to install the Encoder files.
5. To finish the installation, click the **Finish** button.
6. Click the **OK** button.

The following files are installed in the Encoder directory (**Raencode** by default):

File	Description
cevent32	A utility used to create a Synchronized Multimedia presentation.
encoder	The RealAudio Encoder executable.
racut	The Racut tool allows you to cut a piece of a RealAudio file.
rapaste	The Rapaste tool allows you to combine two or more existing RealAudio files that were encoded with the same RealAudio algorithm.
Rax	The Rax tool allows you to modify the text strings or settings of a RealAudio file.
Readme.txt	Information about installation and operation.

Installation on UNIX Platforms

This chapter explains the entire installation and upgrade process for UNIX platforms. If you are upgrading a previous version of the server, read the upgrade procedure described in “Upgrading RealServer from a Previous Version” on page 61 before you begin the installation procedure.

Installing RealPlayer

You will need a Player to test the operation of the server. You can install RealPlayer on the same computer on which you are going to install the server or on any Windows, UNIX, or Macintosh computer that is networked to the Server. The computer on which you install RealPlayer must have a sound card.

To install RealPlayer:

1. With your Web browser, go to the Progressive Networks Web site, which is at the following URL:

<http://www.real.com>

2. Follow the instructions for downloading RealPlayer.
3. Run the setup program that you have downloaded. The setup program installs the RealPlayer software and automatically plays a test clip on RealPlayer. RealPlayer is correctly installed if it plays the test clip.

Installing RealServer

RealServer is available directly from the Progressive Networks site on the World Wide Web. After you fill out the download form, Progressive Networks sends you an e-mail message that contains a customer name and a license key needed to install the server. You will be asked for this information during installation. We recommend that you cut and paste the customer name and license key from the e-mail. This will ensure the information is entered exactly.

RealServer is also available on CD-ROM, which is mailed to you, along with this manual, after you place an order for the server software.

Downloading RealServer From the Internet

To download the RealServer distribution file from the Internet:

1. With your Web browser, go to the URL in the confirmation e-mail you received.
2. Follow the instructions for downloading RealServer. For UNIX, distribution files are compressed files in .tar format. If you download the distribution file to a machine other than the one on which you plan to run it, you must move the distribution file to the correct machine before you uncompress it. Use a utility such as FTP to move the file.
3. Copy the compressed .tar file to a temporary directory and enter the following command:

```
uncompress -c <filename> | tar -xvf -
```

Where <filename> is the name of the downloaded file.

5. Proceed with the installation, as described in the “Running the Setup Program Script” section on page 52.

Installing RealServer from CD-ROM

UNIX-based operating systems require you to mount a new file system or device. The commands needed to mount a CD-ROM differ slightly between these systems.

To mount the CD-ROM:

Sun Solaris

1. Insert the CD-ROM and wait for the operating system to mount the CD-ROM.
2. If you are running File Manager, a window displaying the disk contents appears. If you are not running File Manager, in a shell enter:

```
cd /cdrom/pn_server
```

All Other UNIX-based systems

1. Insert the CD-ROM in the drive.
2. Log in as super-user.
3. From a shell, check whether there is a **/cdrom** directory to mount the CD on; if one does not already exist, enter:

```
mkdir /cdrom
```

4. Enter the appropriate command to mount the CD-ROM:

Operating System	Command
Sun SunOS	<code>mount -rt hsfs /dev/sr0 /cdrom</code>
DEC UNIX	<code>mount -t cdfs -o noversion /dev/rz3c /cdrom</code>
SGI IRIX	<code>mount -rt iso9660 /dev/scsi/sc0d710 /cdrom</code>
IBM AIX	<code>mount -rv cdrfs /dev/cd0 /cdrom</code>

Operating System	Command
Hewlett-Packard HP-UX	<code>mount -rF cdfs /dev/dsk/c0t2d0 /cdrom</code>
FreeBSD	<code>mount -rt cd9660 /dev/cd0a /cdrom</code>
BSD/OS	<code>mount -rt cd9660 /dev/sd1 /cdrom</code>
Linux	<code>mount -rt iso9660 /dev/hdc /cdrom</code>

Note The device name may be different on your computer.

To start the install script:

1. Change directory to the CD-ROM:

```
cd /cdrom
```

Sun Solaris only: Change directory to the **pn_server** directory:

```
cd /cdrom/pn_server
```

2. Change directory to the server directory:

```
cd server
```

HP-UX only: HP-UX computers cannot run **setup.sh** from the CD-ROM. You must copy the **server.tar** file from the CD-ROM to a directory on the computer and untar the file with the following command:

```
tar -xvf server.tar
```

4. Run the setup program which automatically selects the Server version for your platform:

```
./setup
```

Running the Setup Program Script

1. From the directory in which you untarred the distribution file, run the **./setup** program script. The following list of navigational controls are displayed:

<u>Key</u>	<u>Behavior</u>
N	Next
P	Previous
X	Exit
F	Finish (Express Installation)
H	Help

2. Enter the **Customer Name** exactly as provided by Progressive Networks or your Progressive Networks reseller. Use cut and paste if possible to avoid typographical errors. If you enter the wrong customer name, the setup program aborts. Enter [N]ext to continue.
3. Enter the **License Key** exactly as provided by Progressive Networks or your Progressive Networks reseller. Use cut and paste if possible to avoid typographical errors. If you enter the wrong license key, the setup program aborts. Enter [N]ext to continue.
4. The Server License for Progressive Networks RealServer 4.0 is displayed. Read the agreement carefully. Enter [N]ext to accept the terms of the license, or any other key to cancel the installation.

5. The setup program displays information about your server license. The following is sample license information:

```
Your License has the following features enabled:
Valid License
Version:                2
Platform:               NT
Start Date:             5/23/1997
Expiration Date:        5/23/1999
Licensed Streams:       10
User Streams:           0
Multi-Media:            Enabled
Live:                   Enabled
ISP Hosting:             Disabled
ISP Only:               Disabled
User Max Limit:         Disabled
Platform Checking:      Disabled
Intranet Only:          Disabled
Remote License:         Disabled
Remote Streams w/ Local Lic: Disabled
Local Streams:          Enabled
Remote Streams w/ Remote Lic: Disabled
Dynamic ISP licensing:  Disabled
```

6. Enter [N]ext to continue. At this point, all the information necessary for installation has been entered. You may press Express [F]inish to accept the installation default options. Or, you may press [N]ext to set the options yourself.
7. Enter a monitor password, or accept the default (letmein). Enter [N]ext to continue.
8. Enter a Live Encoder password, or accept the default (letmein). Enter [N]ext to continue.
9. Enter a Live File Archive password, or accept the default (letmein). Enter [N]ext to continue.
10. Enter a monitor password, or accept the default (letmein). Enter [N]ext to continue.
11. Enter the TCP port on which the RealServer should listen, or accept the default (7070). Enter [N]ext to continue.
12. Enter the e-mail address to which the RealServer should send system error messages. Enter [N]ext to continue.

13. Enter the e-mail server hostname that RealServer should use to send these e-mails. Enter [N]ext to continue.
14. RealServer 4.0 has the ability to broadcast to players behind firewalls by transmitting all data in HTTP over port 80 after more efficient transports have failed. We strongly recommend that you take full advantage of this feature to maximize the viewership of your content.

If you choose to enable this feature, the RealServer setup program will check port 80 for the presence of a Web server, and warn you if a conflict is found.

```
Would you like to enable Smart Networking on
port 80? (Default: Yes)
```

If a conflict is detected on port 80, and can enable Smart Networking in one of two ways:

- You can shut down the Web server running on the machine running RealServer, and activate Smart Networking with the HTTPPort config file parameter.
- If you have a Web server running on the machine running RealServer, you can configure the machine to have two IP addresses (consult your system documentation for details). Then configure RealServer to use the alternate IP address using the IPBindingList and HTTPPort config file parameters.

4. Enter the user id used to run the server. Enter [N]ext to continue.
5. Enter the group id used to run the server. Enter [N]ext to continue.
6. The RealServer configurations you entered are displayed. Enter [N]ext to begin copying files to your directory.
7. A screen displays information about how to contact Progressive Networks for help

```
Progressive Networks can be contacted as
follows:
```

```
Web:      www.real.com
Tel:      206-674-2700
EMail:    info@prognet.com
```

17. Enter [N]ext to continue. You will be prompted:

Congratulations! RealServer 4.0 has been successfully installed on your computer

18. Enter E[X]it to complete the installation.

Testing RealServer

The RealServer self-test verifies only that RealServer starts and that a connection can be made to RealServer. You should make sure that RealServer is functioning properly by playing a sample media file.

Starting RealServer

If, when you installed RealServer, you specified that RealServer run its self-test, the server should still be running. If you did not run the self-test, you will have to start the server.

To start RealServer:

1. Change to the directory into which you installed RealServer (**pnserver** by default).
2. Start RealServer by entering:

```
bin/pnserver server.cfg
```

RealServer does not return any messages to indicate that it has started.

3. You can determine whether RealServer is running by entering a `ps` command and a `grep` command that searches for **pnserver**—for example,

```
ps -ef | grep pnserver
```

If the server is running, the `ps` and `grep` command return two lines for the `pnserver` processes that look something like this:

```
username 25851 25850 0 16:51:11 0:00 bin/pnserver server.cfg
username 25850      1 0 16:51:11 0:00 bin/pnserver server.cfg
```

The above lines indicate the `pnserver` process and the `pnserver` resolver process (which is a child process of the `pnserver` process) are running. The PIDs for the two processes should be sequential. In the above example, the `pnserver` process has a PID of 25850, and the `pnserver` resolver process has a PID of 25851.

If RealServer does not start, review the error messages in the server error log (**`pnserver/logs/pnerror`** by default).

Playing a Sample Clip

Once RealServer starts, you can test it by playing a clip in the **content** directory.

To play a sample clip:

1. Start RealPlayer on any Macintosh, Windows or UNIX computer that can access your RealServer through a network.
2. From the RealPlayer File menu, select **Open Location**.
3. In the **URL** box, enter the path of the media file—for example:

```
pnm: //<my.pnserver>:<port>/welcome.rm
```

where:

<my.pnserver> is the DNS name or the IP address of the computer with RealServer installed

<port> is the TCP/IP port (7070 by default) on which RealServer listens for client connections

If RealPlayer plays the media file, then RealServer is installed correctly.

If the media file does not play at all, or if the performance or media quality is poor, see the Troubleshooting chapter for information to help you diagnose and correct the problem. Also check your log files for clues. To learn about log files, see “RealServer Access and Error Logs” on page 138.

Note By default, media files are located in the **pnserver/content** directory. In general, you should not put media files in directories below your Web server. RealServer is independent of and does not communicate with the Web server.

Stopping RealServer

To stop RealServer:

1. Log on either as super-user or as the user (UID) which RealServer runs as.
2. If you know the current process ID for RealServer, enter:

```
kill <processid>
```

3. If you don't know RealServer's process ID, change to the **pnserver** directory and enter:

```
kill `cat logs/pnserver.pid`
```

Note For information on gracefully shutting down RealServer when users are connected to the server, see “Shutting Down Gracefully” on page 158.

RealServer Directories and Files

The RealServer setup program script creates and installs the following directories and files **pnserver** is the default directory for the server files. The following directories are created below the server directory:

- **bin**
- **docs**
- **logs**
- **content**

The pnserver Directory

The **pnserver** directory contains the following ASCII text files:

File	Description
server.cfg	The server configuration file, which contains the configuration settings for the server.

The bin Directory

The **bin** directory contains the following executable files:

File	Description
pnserver	The RealServer executable.
rssm	The system manager executable.
rmfile	The live RealMedia file creation utility, which creates (archives files from live broadcasts.
rafile	The live RealAudio file creation utility, which creates (archives) files from live broadcasts.
rmcut	Cuts a specific portion of a RealVideo file without changing the original source.
rpmaste	Combines two or more RealVideo files.
rmdump	Outputs the contents of an rmfile to a text file.
rmedit	Modifies the title, author, copyright, comment, mime type, or stream name. It can also be used to print the current values for the file or stream.

The docs Directory

The **docs** directory contains the following ASCII files:

File	Description
PROBLEM	A list of known server problems.
README	Information about server installation and operation.
License.txt	Server license agreement.

The logs Directory

The **logs** directory contains the following log files, which are in ASCII text format:

File	Description
pnaccess	The access log, which logs information about clients that have connected to the server.
pnerror	The error log, which logs information and error messages about server operation.
pnserver.pid	When the server is running, this file contains the server's current process ID (PID) When the server is not running, this file contains the server's last-used PID.

The Content Directory

The **content** directory is installed with sample media files. The content directory is normally where you should store the media files delivered by RealServer. The BasePath configuration setting (see page 79), which specifies the root directory of your media files, points to **/pnserver/content** by default.

Troubleshooting RealServer Installation

This section describes the most commonly encountered installation problems. This section also describes how to recover from the unlikely event of a RealServer crash and how to determine whether RealServer has restarted from a crash.

Incorrectly Entered Customer Name or License Key

If you enter the wrong customer name or license key, the program script aborts. If you typed in the customer name or license key, check whether you made any typing errors. If you pasted in the customer name and or license key from the e-mail message that you received from Progressive Networks, make sure that you pasted in the entire name and license key and that you did not introduce any extra spaces.

If you have verified that the entered customer name and license key are correct, but they still do not work, contact technical support at Progressive Networks.

The Assigned User Name or Group is Invalid

If you assign a user name (UID) to the server, you must assign a user name that is defined on the computer. If you assign a group (GID) to the server, you must assign a group that is defined on the computer. If you assign a UID or GID that does not exist on the computer, RealServer will not start.

Recovering from Server Crashes

In the unlikely event that the server crashes, use the `ps` command to display the process ID (PID) of the `pnserver` and `pnserver resolver` processes—for example:

```
ps -ef | grep pnserver
```

If the server is running, the `ps` and `grep` commands will return two lines for the `pnserver` processes that look something like this:

```
username 25851 25850 0 16:51:11 0:00 bin/pnserver server.cfg
username 25850      1 0 16:51:11 0:00 bin/pnserver server.cfg
```

The above lines indicate the `pnserver` process and the `pnserver` resolver process (which is a child process of the `pnserver` process) are running. The PIDs for the two processes should be sequential. In the above example, the `pnserver` process has a PID of 25850, and the `pnserver` resolver process has a PID of 25851.

If the server has crashed, the `pnserver` process usually has stopped but the `pnserver` resolver process, which is a child process of the `pnserver` process, is usually still running. Kill the `pnserver` resolver process and restart `RealServer`.

How to Verify that RealServer has Restarted

The `pnserver/logs/pnserver.pid` file logs the current process ID (PID) for `RealServer`. If you want to verify that `RealServer` has in fact restarted, you can check the process ID contained in this file and then attempt to restart the server. Then check the `pnserver.pid` file again. If the PID is different than the PID you recorded earlier, the server has restarted. If the PID is not different, the server did not restart.

Upgrading RealServer from a Previous Version

If you are upgrading from a previous version of `RealServer`, you should install the new version next to, and not on top of, the old version. Once you have installed and tested the new `RealServer`, you can then replace your old version.

Installing the New Version

To install without replacing an existing RealServer:

1. Install the new version in a different directory than your existing `RealServer`. This prevents you from affecting your existing `RealServer`.
2. Note the `PnaPort` entry in your existing `RealServer` configuration file (`/pnserver/server.cfg` by default). If the `PnaPort` entry is set to 7070 or is not present, `RealServer` is using the default port 7070. You need to use a

different port number for testing the newly installed RealServer. To use a different port number, add or edit the following line to your **server.cfg** file after installing:

```
PnaPort      7072
```

Be sure to specify a port that is not used by another configuration parameter such as **ResolverPort**. If you have problems starting RealServer on this port because another application is using it, try using a different port number.

3. To send a test URL to the new RealServer, you must add :7072 to the pnm URL. This makes the test URL become:

```
pnm://<my.server>:7072/144.ra
```

Configuring the New Version

After you have tested your installation of RealServer, you can then have it duplicate serving of your existing content.

1. Copy the **BasePath** entry from the **server.cfg** file of your existing RealServer to the **server.cfg** file for the new version. It is best to use an absolute base path for the **BasePath** entry.
2. Send a test URL to the new Server. Remember to add :7072 to the pnm URL. This makes the test URL become:

```
pnm://<my.server>:7072/144.ra
```

Moving the New Version into Production

When you are satisfied the newly installed RealServer is supplying your existing content, you can stop your old server, and move the new server into production.

To move the new version into production:

1. Stop both instances of RealServer.

2. Rename the directory containing the old installation.
3. Rename the directory containing the newly installed RealServer, using the original name of the old server software.
4. Merge all appropriate settings from your existing configuration file into your new configuration file.
5. Set the **PnaPort** entry back to its original value.
6. Change your Web page links back to their original port number.

Starting and Testing the System Manager

The System Manager executable, **rssm**, is installed in the **pnservice/bin** directory when you install RealServer. With the System Manager, you can monitor the server's performance, check and modify the server configuration, view a list of clients that are connected to the server, and perform other administrative activities.

Before you can start the System Manager, you must use the UNIX **chmod** command to assign execute privileges to **rssm**.

You can then test System Manager by connecting to a RealServer that is running. For example, the command:

```
rssm -i -p password myserver:7070
```

connects System Manager in interactive mode through TCP/IP port 7070 to a RealServer running on computer **myserver**.

See page 173 for a complete description of the **rssm** command. See page 107 for a description of how to use System Manager to monitor RealServer.

Installing RealAudio Encoder

You can download RealAudio Encoder from the Progressive Networks Web site or you can install it from the RealServer CD-ROM.

To download the Encoder from the Internet:

1. Using a Web browser, download the Encoder from:

`http://www.real.com`

2. When prompted, click the **Save to Disk** button and specify the temporary directory for the Encoder installation program.
3. Change directories to the directory that contains the downloaded file. Decompress and untar the file by entering:

```
uncompress <filename> | tar -xvf -
```

where <filename> is the name of the downloaded file.

Untarring the file creates the **raenc** directory and installs the encoder files in this directory.

To install the Encoder from the CD-ROM:

1. Mount the CD-ROM drive See “Installing RealServer from CD-ROM” on page 50 for more information.
2. Copy the correct distribution file for your platform from the **Encoders/Audio** directory of the RealServer CD-ROM.
3. Change directories to the directory that contains the distribution file and untar the file by entering:

```
tar xfv <filename>
```

where <filename> is the name of the install file.

Untarring the file creates the **raenc** directory and installs the RealAudio Encoder files in this directory.

To add the Encoder to your path:

So that you can start the RealAudio Encoder from any directory on the computer, add the programs to your path :

- If you are using the csh, zsh, or tsch shell, add the following command to your .cshrc file after all other “set path” commands:
`set path=($path $HOME/<your_encoder_dir_name>)`
- If you are using the ksh, bash, or bsh shell, add the following command to your profile file after all other PATH commands:
`PATH=$PATH:$HOME/<your_encoder_dir_name>`

To start the RealAudio Encoder:

- Enter the following command:

```
raencoder
```

The Encoder prints usage information.

RealAudio Encoder Directory and Files

The following files are installed in the **raenc** directory:

File	Description
license	An ASCII text file that contains the license agreement for the Encoder.
readme.enc	Information about Encoder installation and operation.
cevents	The synchronized multimedia compiler.
encoder.cfg	The configuration file for the encoder You can store operation parameters in this ASCII text file rather than specifying them as command-line options when starting the Encoder.
racut	The Racut tool allows you to cut a piece of a RealAudio file.

File	Description
racut.man	This man page provides information on the Racut tool and the Rapaste tool.
raencoder	The RealAudio Encoder executable.
raencoder.man	This man page provides information on the RealAudio Encoder.
rapaste	The Rapaste tool allows you to combine two or more existing RealAudio files that were encoded with the same RealAudio algorithm.
rax	The Rax tool allows you to modify the text strings or settings of a RealAudio file.
rax.man	This man page provides information on the Rax tool.

Configuration File

This chapter introduces the server configuration file (**server.cfg**), which contains all the configuration settings for RealServer. When you want to configure an aspect of server operation, you must edit the configuration file. This chapter also describes how to edit the configuration file with a text editor or with the System Manager.

As part of the installation procedure, the server installer, on both Windows NT and UNIX, generate a default server configuration file, **server.cfg**. This is a plain-text file located in the server directory (**pnserver** by default) on UNIX, and in the **Real\Server** directory on Windows NT. The configuration file contains all the configuration settings for RealServer. You may add or modify parameters by editing the configuration file with your text editor.

The configuration file stores pairs of configuration options and their settings, such as:

```
LocalHost           Matisse
LiveFilePassword    fauvist
EncoderTimeout 30
```

Note that entries in the file are case-sensitive.

Changes to some of the configuration values have an immediate effect; others take effect only after you restart RealServer. For specific information about individual options, including when changes to a particular option take effect, see the following chapters: "Configuring Basic Server Settings," "Configuring Advanced Server Settings," "Live Broadcasting," and "Multicasting".

Sample Configuration File for Windows NT Installations

The following is a sample configuration file for Windows NT installations.

```
CustomerName rosebud
LicenseKey
57839283948n83928h092929j293hs309a93029j209a0920358103902
9a91093909309

BasePath C:\Real\Server\Content
LogPath C:\Real\Server\Logs\Pnaccess.log
ErrorLogPath C:\Real\Server\Logs\Pnerror.log

PnaPort 7070

AudioConnections 10

MonitorPassword letmein

EncoderPassword letmein

LiveFilePassword letmein

HTTPPort 80
```

Sample Configuration File for UNIX Installations

The following is a sample configuration file for UNIX installations:

```
#Stats mask (3 enables complete stats logging)
StatsMask 3

#Mail message user
#MailMessageUser your_pop_login

#Mail host
#MailMessageSMTPHost your.mailhost.com

#Mail message limit
MailMessageLimit 2

#Mail message level
MailMessageLevel INFO

#Mail usage CC
#MailUsageCC auto-feedback@prognnet.com

#Mail usage threshold (percent of server capacity)
#MailUsageThreshold 90

#Mail usage summary period (hours)
#MailUsagePeriod 24

#Base path for URLs.
BasePath /export/home/jaisimha/src/tesst/installtest/content

#Filename (not just directory) to which log entries should
be appended.
LogPath
/export/home/jaisimha/src/tesst/installtest/logs/pnaccess.log

#Filename to which error log entries should be appended.
ErrorLogPath
/export/home/jaisimha/src/tesst/installtest/logs/pnerror.log
#Filename holding pnserver process ID
PidPath
/export/home/jaisimha/src/tesst/installtest/logs/pnserver.pid

#7070 is the standard port number.
PnaPort 7070

#Port for Smart Networking
HTTTPort 80

#Set maximum number of simultaneous System Manager
connections
MonitorConnections 2

#Set timeout for idle connections
Timeout 180
```

Editing the Configuration File

There are two ways to edit the configuration file: with a text editor or with the System Manager. With a text editor, you can edit the file whether the server is running or not. You can edit the configuration file with the System Manager only when the server is running.

Using a Text Editor to Edit the Configuration File

You can edit the configuration file with any text editor. When RealServer is not running, editing the configuration file with a text editor is the only way to change settings.

On Windows NT Platforms

1. Using a text editor such as Notebook or Wordpad, open the **server.cfg** file located in the **Real\Server** directory.
2. Edit the entry or entries you want to change.

For specific information about individual options, including when changes to a particular option take effect, see the following chapters: "Configuring Basic Server Settings," "Configuring Advanced Server Settings," "Live Broadcasting," and "Multicasting".

3. Save the file.
4. Depending on the configuration settings that you have changed, you may have to restart the server for the changes to take effect.

Note The configuration file must be saved in a text only format and no line feeds should be included in the file.

On UNIX Platforms

1. Using a text editor such as vi, open the **server.cfg** file located in the **pnserver** directory.
2. Edit the entry or entries you want to change.

For specific information about individual options, including when changes to a particular option take effect, see the following chapters: "Configuring Basic Server Settings," "Configuring Advanced Server Settings," "Live Broadcasting," and "Multicasting".

3. Save the file as text.
4. To force reloading of the new configuration settings, use the command:

```
kill -HUP `cat pnserver.pid`
```

For information on using the `kill` command, see page 166.

Using the System Manager to Edit the Configuration File

When RealServer is running, you can change its configuration with System Manager. You can use System Manager to reconfigure a RealServer running on any platform, as long as you can connect System Manager to RealServer over a network.

You can run multiple sessions of System Manager simultaneously. If you have more than one RealServer running, you can start a session for each one and leave them all running continuously for as long as RealServer is running. You can also run multiple sessions for any particular RealServer.

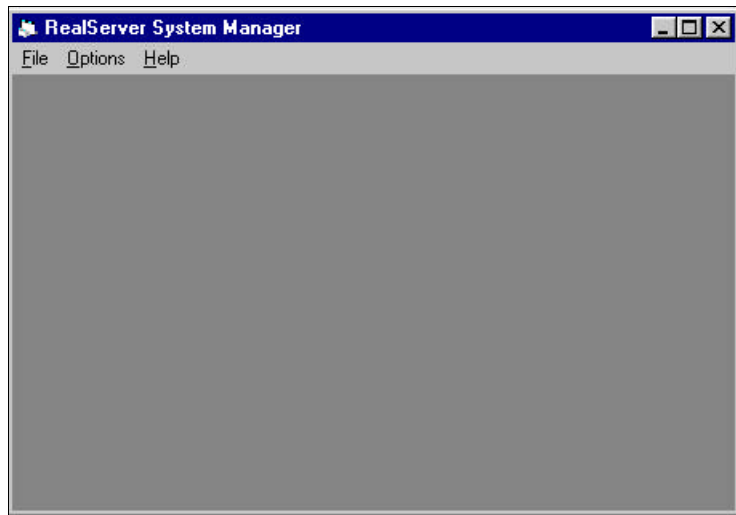
RealServer System Manager running on any platform can configure a server on any other platform. For example, a Windows NT RealServer System Manager can configure an IRIX server.

From Windows 95 or Windows NT Platforms

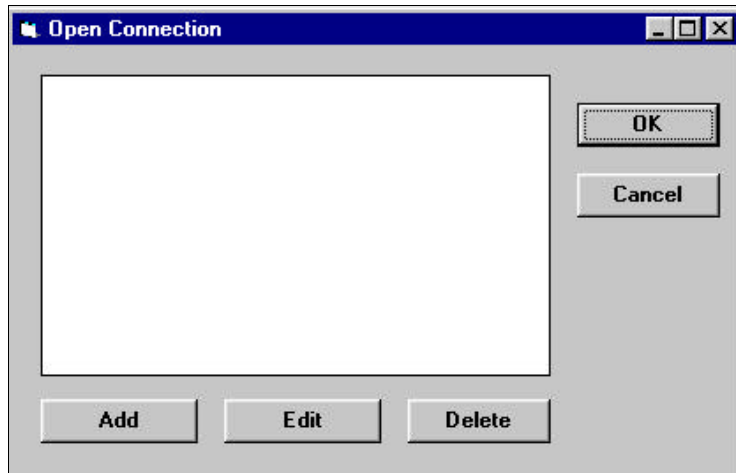
To connect System Manager to RealServer:

1. **Windows 95 and NT 4.0:** Click the **Start** button, point to **Programs**, point to **RealServer**, and click **System Manager**.

Windows NT 3.5: Double-click the RealServer program group and double-click the **System Manager** icon.



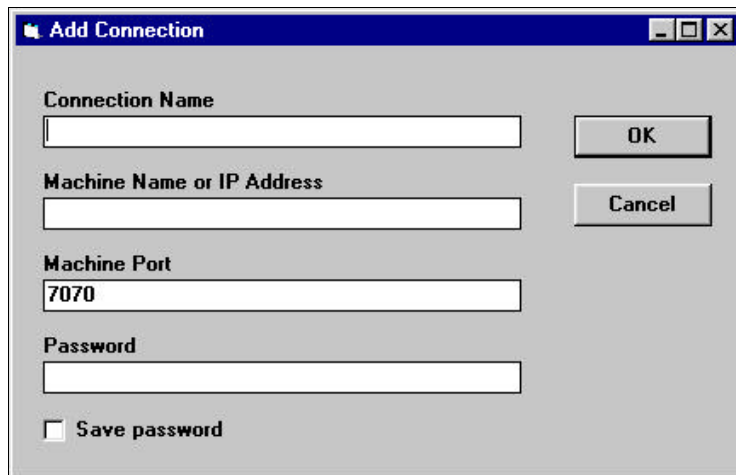
2. On the File menu, click **Open**.



3. Click the name of the RealServer that you want to monitor.
4. Click the **OK** button.

If the RealServer you want to monitor is not in the list, you need to add it to System Manager's selection menu.

1. In the **Open Connection** dialog box, click the **Add** button.



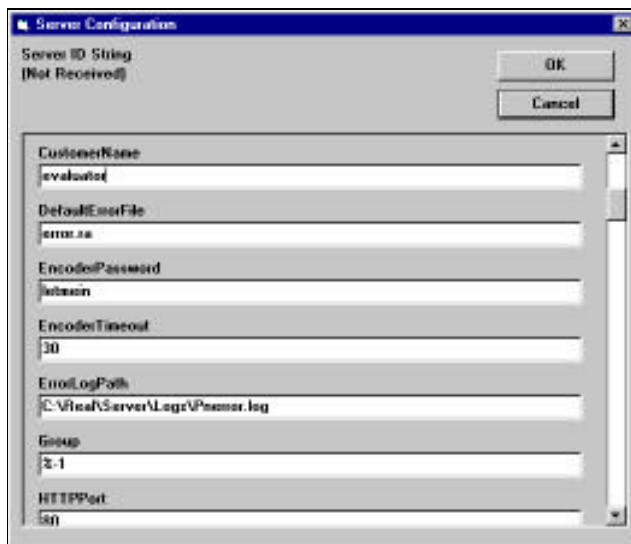
2. Enter the required information and click the **OK** button:

Text box	Enter
Connection Name	A name for the connection.
Machine Name	The name of the machine running the Server you want to reconfigure.
Machine Port	The port number for System Manager to use to reach RealServer. This is the port specified in the PnaPort setting for RealServer (the default is 7070).
Password	The password required to connect System Manager to RealServer. (See MonitorPassword on page 108.

System Manager automatically saves the connection information for all of the RealServers you enter.

To change configuration options:

1. On the Server menu, click **Configuration**.



2. Edit the entry or entries you want to change and click the **OK** button.

All configuration options are shown, even if some of them do not have a value.

From UNIX Platforms

The System Manager is installed as part of the RealServer distribution on UNIX platforms. The System Manager executable file is named **rssm** and is located in the **pnserver/bin** directory.

To connect System Manager to RealServer:

1. Start the System Manager in interactive mode. (For a complete description of the **rssm** command, see page 173). For example, to start the System Manager from the **pnserver/bin** directory, enter:

```
./rssm -i -p password host:port
```

where:

<password> is the password for the System Manager (see MonitorPassword on page 108)

<host> is the name or IP address of the computer on which RealServer is running.

<port> is the TCP/IP port on which RealServer is listening for client connections (see PnaPort on page 85).

2. The System Manager displays the interactive command prompt:

Command :

To display and change configuration settings:

1. To display all the configuration settings in the configuration file, enter **c**
2. To change a configuration, enter **n**

3. Enter the name of the configuration setting that you want to change. For example, if you want to change the TCP/IP port on which RealServer listens for client connections, enter **PnaPort**:

Enter config var name: **PnaPort**

4. The System Manager indicates the data type for the setting and the current value of the setting. The System Manager then prompts you for a new value. For example:

This var is an integer. Its current value is: 7070.
Please enter new value for 'PnaPort':

5. Enter the new value. For example, if you want to change the PnaPort to 7073, enter **7073**

6. The System Manager indicates that the value of the configuration setting has been changed and prompts you to confirm the change. Confirm the change by entering **c**. For example:

'pnaport' has been set. Press 'c'
to make sure it has been set correctly.

Command: **c**

7. The System Manager displays all of the configuration settings in the configuration file. Make sure that the configuration setting you changed has the value you want.

8. To exit System Manager, enter **x**.

Configuring Basic Server Settings

This section describes the settings in the server configuration file (**server.cfg**) that control the basic operation of RealServer. You do not require specially licensed features to use these settings.

The basic configuration settings can be divided into the following groups:

- **General settings**—control miscellaneous aspects of the server operation.
- **Logging settings**—specify the information that the server logs and where this information is logged.
- **E-mail settings**—configure e-mail notification from the server, including certain server operation thresholds that generate e-mail messages from the server when exceeded.
- **Bandwidth and Connection settings**—limit the bandwidth used by the server and the number of simultaneous client connections to the server.
- **Security setting**—specifies the network domains that can access RealServer.
- **System Monitor settings**—control System Manager's access to the server.

Note After installing RealServer, you do not have to change any of the basic settings for RealServer to operate properly. RealServer is installed with default configuration settings that should work on your computer. You may find it useful, however, to customize RealServer for your particular requirements and environment.

General Settings

This table summarizes the general configuration settings that control miscellaneous aspects of RealServer operation. A detailed description of each setting follows the table.

Setting	Description	Restart Server After Changing
BasePath	Specifies the path to the root directory of your media files.	No
CustomerName	Your customer name, as received from Progressive Networks. Must be correct for the server to operate.	No
Group (UNIX only)	Default group ID (GID) for server.	Yes
LicenseKey	Encrypted license string from Progressive Networks. Must be correct for the server to operate.	No
LocalHost	Fully qualified domain name that overrides the system default name.	No
MaxThreads	Limits the number of threads used by the server.	Yes
MinPlayerProtocol	Specifies the minimum PN protocol that a Player must have to connect to the server.	No
PnaPort	Specifies the TCP/IP port on which the server listens for connection requests from clients.	Yes

Setting	Description	Restart Server After Changing
PIDPath (UNIX only)	Specifies the file used to record the server's process ID.	No
Timeout	Specifies the number of seconds the server waits before disconnecting an inactive Player.	No
User (UNIX only)	The default user ID (UID) for the server.	Yes
HTTPPort	Enable players that may be behind firewalls to access content served from your server.	Yes
ResolverPort	The TCP port to use for resolving DNS addresses.	No

BasePath

Path to root directory of your media files.

Used by	pnserver
Default value	content
Range of values	Valid directory names
Restart Server after change	No

Most media content delivered by your RealServer resides in, or beneath, the directory specified by the base path. Use the following format to enter the absolute path name of the base path directory into the **server.cfg** file:

BasePath <path>

BasePath should point to the **content** subdirectory of your RealServer installation directory, since this directory contains several sample documents. For further information on organizing media content see “Bandwidth Negotiation” on page 271 in the RealAudio Content Creation Guide and on page 223 in the RealVideo Content Creation Guide sections of this manual.

Example

BasePath /pnserver/content

CustomerName

Your name specified in the license you received from Progressive Networks or from your Progressive Networks reseller (by e-mail).

Used by	pnserver
Default value	(none)
Range of values	Valid alpha-numeric string
Restart Server after change	No

For your RealServer to function, you must enter this parameter and the **LicenseKey** value. You must enter the exact customer name (as well as the exact license key) or RealServer will not operate.

CustomerName <licensename>

If you purchase a new or upgraded license, you must enter the new CustomerName and LicenseKey values that you receive from Progressive Networks or from your Progressive Networks reseller.

Example

CustomerName WorldJam Corporation

DefaultErrorFile

File to play when a media file requested by a Player is inaccessible.

Used by	pnserver
Default value	error.ra
Range of values	Media file name and path
Restart Server after change	No

RealServer sends a Player an error message when a requested file is not available. If you set the **DefaultErrorFile** setting, RealServer plays the specified media file instead of sending the error message.

```
DefaultErrorFile    <path>
```

The path to your error file should be an absolute path. Your error file should be an audio file recorded in 14.4 format and indicate that there was a format compatibility problem. For example, “We are sorry but the file requested is not available in your Player’s format. Please try another file.”

Note You can specify a video (.rm) file for **DefaultErrorFile** rather than an audio (.ra) file. However, since RealAudio Players 3.0 (or earlier) will not be able to play a video file, this is not recommended.

Example

```
DefaultErrorFile    pnserver/content/nofile.ra
```

Group (UNIX only)

Default group name for RealServer for UNIX.

Used by	pnserver
Default value	(none)
Range of values	Valid user and group names
Restart Server after change	Yes

The group name must exist on the computer on which RealServer is running; otherwise, RealServer will not start.

If you do not specify a group name when installing RealServer, the group name defaults to the group name of the user who first starts RealServer.

Example

Group users

LicenseKey

Encrypted license string enabling your RealServer to operate.

Used by	pnservice
Default value	Default license allows two streams
Range of values	N/A
Restart Server after change	No

For your RealServer to function, you must enter this setting and the **CustomerName** setting. You must enter the exact license key and customer name or RealServer will not operate.

LicenseKey <encryptedkey>

If you downloaded your software, `encryptedkey` is in the e-mail that gives you access to the download URL. Use a word processor or text editor to cut and paste the license information. If you purchased your software on CD-ROM, `encryptedkey` is provided through e-mail from Progressive Networks or is affixed to your CD-ROM case or RealServer software package.

If you purchase a new or upgraded license, you must enter the new CustomerName and LicenseKey values that you receive from Progressive Networks or from your Progressive Networks reseller.

Example

LicenseKey 43819m554420998372983729857298752983
 758hf2938299192384j6esu3829879298

LocalHost

Fully-qualified domain name that overrides the system default domain name.

Used by	pnserver
Default value	(none)
Range of values	valid domain name
Restart Server after change	no

On some platforms, the system does not return a fully-qualified domain name, which causes difficulty for RealServer in locating other RealServers in a multiprocessing configuration. With the **LocalHost** parameter, you can override the system default domain name and provide RealServer with a fully-qualified domain name:

```
LocalHost      <domain name>
```

If you experience problems running multiple processes, you can set the **LocalHost** parameter in your RealServer configuration file.

Example

```
LocalHost      mycomputer.mydomain.com
```

MaxThreads

Maximum number of threads, or processes, used by RealServer.

Used by	pnserver
Default value	1
Range of values	Integers greater than zero
Restart Server after change	Yes

This entry lets RealServer run multiple processes within a single machine. RealServer can take advantage of multiple CPU's.

```
MaxThreads          <count>
```

Note This configuration parameter affects your computer's CPU usage.

Example

```
MaxThreads          5
```

MinPlayerProtocol

The minimum PN protocol supported by RealServer.

Used by	pnserver
Default value	0
Range of values	0, 4, 7, 9, 10
Restart Server after change	No

Players that do not supply a protocol number equal to or greater than this value as part of their connection information cannot connect to RealServer.

```
MinPlayerProtocol <number>
```

Valid values are:

- 0** All players
- 4** RealAudio Player 1.0 and later (same as 0)
- 7** RealAudio Player 2.0 and later
- 9** RealAudio Player 3.0 only
- 10** RealPlayer only

Example

To allow only RealAudio 2.0 and later players, enter:

```
MinPlayerProtocol 7
```

PnaPort

Number of the TCP port RealServer uses for receiving requests from clients.

Used by	pnservice
Default value	7070
Range of values	Valid port number
Restart Server after change	Yes

```
PnaPort <number>
```

To use a port lower than 1024 on a UNIX system, you need to be logged on with super-user privileges. The only reason to use a port other than the default is to allow several Servers to coexist on one system, or to achieve some level of privacy when serving information by using an unusual port number.

Example

```
PnaPort 7074
```

PidPath (UNIX Only)

File used by RealServer for UNIX to record its process ID.

Used by	pnserver
Default value	pnserver.pid
Range of values	Valid file name
Restart Server after change	No

If you do not specify a PidPath, RealServer for UNIX records its process ID in **pnserver/logs/pnserver.pid**. Use the following format to specify a different path or filename for the process ID log:

```
PidPath <path>
```

For simple administration, the process ID file should reside in the same directory as your access and error log files.

Example

```
PidPath pnserver/logs/pnserver.pid
```

Timeout

Number of seconds RealServer waits before disconnecting an inactive Player.

Used by	pnserver
Default value	300 seconds
Range of values	120 - 900
Restart Server after change	No

Because every connection consumes valuable resources, connections should not be permitted to sit idle for long periods of time. A connection is idle when the Player has paused playing of media or has reached the end of the media program without disconnecting.

Timeout <Seconds>

The client can automatically reconnect after being timed out by RealServer if the user clicks the **Play** button.

Example

Timeout 240

HTTPPort

Enable players that may be behind firewalls to access content served from your server.

Used by	pnserver
Default value	If not set, the port is not opened and HTTP cloaking is not available.
Range of values	Should always be set to 80
Restart Server after change	Yes

ResolverPort

The TCP port to use for resolving DNS addresses.

Used by	pnservice
Default value	PnaPort + 1
Range of values	Valid port number
Restart Server after change	No

```
ResolverPort <port>
```

Example

```
ResolverPort 8081
```

User (UNIX only)

Default user name for RealServer for UNIX.

Used by	pnservice
Default value	(none)
Range of values	Valid user and group names
Restart Server after change	Yes

```
User <UserName>
```

The user name must exist on the computer on which RealServer is running; otherwise, RealServer will not start.

If you do not specify a user name when installing RealServer, the user name defaults to the user name of the user who first starts RealServer.

Example

```
User fredk
```


Configuring RealServer for Smart Networking

Smart Networking allows RealServer to stream content to a RealPlayer behind a firewall that does not permit TCP/UDP transmissions. This is done through a HTTP-like protocol.

No Web Server

If there is no Web server running on the machine running RealServer, the server can be configured to perform HTTP transmission on port 80. To do this, enter the form:

HTTPPort 80

in your server.cfg file. Although you can use any valid TCP/IP port number, RealPlayer will try to connect to port 80. If port 80 is not available, the server will not perform HTTP transmissions.

Note To use port 80 on UNIX, the server must be started as root (super-user).

Setting	Description	Restart Server After Changing
HTTP	Sets the incoming connection port for HTTP requests to RealServer. If not specified, no HTTP connections are possible. The installer will default to port 80 if no other server is detected on that port.	Yes

Web Server

If there is a Web server on port 80 of the machine hosting RealServer, you can move the Web server to another machine, move the Web server to another port, or configure the server to transmit on an alternate IP address. RealServer provides support for binding to multiple IP addresses. This allows the server to

be configured to transmit on multiple IP addresses and/or use multiple network interfaces. To do this, enter the form:

```
IPBindingList [add1, add2, ...]
```

where: add1, add2 are valid IP addresses that have been obtained and associated with the host machine.

Note You must put commas between each IP address in the list. If a HTTPPort entry is also present in the server.cfg file, then the designated port is opened on each IP address and can be used as a mechanism for Smart Networking.

Setting	Description	Restart Server After Changing
IPBindingList	Specifies the list of valid IP addresses the server can bind. If not specified, then all addresses are bound.	Yes

To set up virtual IP addresses on Windows NT:

1. From Settings, select **Control Panel**.
2. Double-click the **Network** icon.
3. In the Protocol tab, select TCP/IP, and click **Properties**.
4. From this point, there are two possible scenarios:
 - If you have two ethernet cards, you can select the other card. Change the last digits of the IP address.
 - If you have one ethernet card, click **Advanced**, then click the **Add** button. Type in the new IP address. Click the **Add** button.
5. Reboot your system.

To set up virtual IP addresses on UNIX:

The procedure for setting up virtual IP addresses on UNIX depends upon the operating system. Refer to your operating system manual for more

information. If your system does not automatically route traffic to the new address, be sure to manually route to the virtual IP address.

Configuring Access and Error Logging

You can specify, to a certain degree, the information that RealServer logs about client connections and errors. You can also specify where RealServer logs this information.

The following table summarizes the access and error log settings. A detailed description of each setting follows the table.

Setting	Description	Restart Server After Changing
ErrorLogPath	Specifies the path and file name of the error log file.	No
LogPath	Specifies the path and file name of the access log file, which logs information about client access to RealServer.	No
LoggingStyle	Specifies whether to use the original access log format or the new format with additional information. There are three modes: 0 1 (RealAudio 3.0) 2 (RealServer 4.0).	No
StatsMask	Specifies additional access log statistics to request from Players.	No

ErrorLogPath

File storing information about errors which occur during the operation of RealServer.

Used by	pnserver
Default value	pnerror.log
Range of values	Valid file name
Restart Server after change	No

Enter a line using the following format into **server.cfg**:

```
ErrorLogPath <path>
```

During installation, ErrorLogPath is set to the file **pnerror.log** in the **logs** subdirectory of your RealServer installation directory. If the ErrorLogPath setting is not present in the **server.cfg**, RealServer records errors in the **pnerror.log** file located in the same directory as **pnserver**.

Example

```
ErrorLogPath    pnserver/pnerror1.log
```

LogPath

Path and file name of the access log file.

Used by	pnserver
Default value	pnaccess.log
Range of values	Valid filename
Restart Server after change	No

RealServer logs information regarding every access to your Server into the file specified by the **LogPath**. Enter a line using the following format into the **server.cfg**:

LogPath <path>

During installation, this LogPath is set to the file **pnaccess.log** in the **logs** subdirectory of your RealServer installation directory. If the LogPath setting is not present in **server.cfg**, RealServer records errors in the **pnaccess.log** file located in the same directory as **pnserver**.

Example

LogPath logs/pnaccess.log

Uses a relative path from the directory from which RealServer was started.

LoggingStyle

Specifies whether to use the original access log format or the new format with additional information.

The **StatsMask** configuration parameter specifies which additional information is included when **LoggingStyle** is set to 1 or 2. Style 2 adds a unique player ID to style 1.

Used by	pnserver
Default value	0
Range of values	0, 1, 2
Restart Server after change	No

LoggingStyle <value>

To include the additional information in the access log file, set value to 1. For more information about log files, see “Using the Access and Error Log Files” on page 136 and “Access and Error Log Messages” on page 138.

Example

LoggingStyle 1

StatsMask

Specifies additional access log statistics to request from Players.

These statistics are included in the access log only when the **LoggingStyle** configuration parameter is set to 1.

Used by	pnservice
Default value	0
Range of values	0 - 3
Restart Server after change	No

```
StatsMask <value>
```

Where <value> is:

- 0** No additional statistics
- 1** Statistics type 1 only
- 2** Statistics type 2 only
- 3** Both statistics types 1 and 2

Note Statistics type 2 are returned only by RealAudio Player 3.0.

For more information about access log files, see “Using the Access and Error Log Files” on page 136 and “Access and Error Log Messages” on page 138.

Example

```
StatsMask 3
```

Configuring E-mail and Usage Thresholds

You can configure RealServer to send e-mail messages that alert you to important RealServer events. You can specify the following aspects of e-mail operation:

- One or two addresses to which RealServer sends e-mail messages
- The SMTP mail server that RealServer uses to send e-mail messages
- The types of messages (informational, warning, or error) that RealServer sends
- The number of times RealServer sends an e-mail message about a particular event (such as a usage threshold being crossed)
- A RealServer usage threshold that, when exceeded, causes RealServer to notify you by e-mail

Configuring Basic E-mail Operation

The following configuration settings control the basic aspects of RealServer e-mail operation.

Setting	Description	Restart Server After Changing
MailMessageLevel	Specifies the severity level of e-mail messages sent by RealServer.	No
MailMessageLimit	Limits the number of e-mail messages RealServer sends about a particular event.	No
MailMessageSMTPHost	Specifies the mail server that RealServer uses to send e-mail messages.	No

Setting	Description	Restart Server After Changing
MailMessageUser	Specifies the e-mail address of the primary person to receive RealServer e-mail messages.	No
MailUsageCC	Specifies a secondary address to which RealServer sends e-mails.	No

MailMessageLevel

Specifies the severity of messages that are e-mailed to the system administrator.

Used by	pnservice
Default value	(none)
Range of values	INFO, WARNING, ERROR
Restart Server after change	No

`MailMessageLevel <level>`

where `level` can be INFO, WARNING, or ERROR, in ascending order of severity. Specifying a level prevents RealServer from sending lower-level messages. The default value, which is no value, prevents RealServer from sending any e-mail messages.

Example

`MailMessageLevel WARNING`

Sends e-mail about warning and error message, but not informational messages.

MailMessageLimit

The number of times that RealServer sends a specific e-mail message.

Used by	pnservice
Default value	5
Range of values	Integers greater than or equal to 0
Restart Server after change	No

`MailMessageLimit <number>`

Use this setting to limit the number of times RealServer notifies the system administrator of the same problem.

You can disable e-mail messages from RealServer by setting MailMessage Limit to 0.

Example

```
MailMessageLimit 3
```

MailMessageSMTPHost

The e-mail server that RealServer uses to send e-mail messages.

Used by	pnservice
Default value	(none)
Range of values	Valid Domain Name System (DNS) name or IP address
Restart Server after change	No

```
MailMessageSMTPHost <address>
```

Example

```
MailMessageSMTPHost mail.mycorp.com
```

MailMessageUser

The e-mail address of the system administrator to receive e-mail messages from RealServer.

Used by	pnservice
Default value	(none)
Range of values	Valid e-mail address
Restart Server after change	No

```
MailMessageUser <address>
```

Example

```
MailMessageUser sysadmin@mycorp.com
```

MailUsageCC

The e-mail address, in addition to that specified in **MailMessageUser**, to receive e-mail from RealServer.

Used by	pnservice
Default value	sales@prognetwork.com
Range of values	Valid e-mail address
Restart Server after change	No

MailUsageCC <address>

Example

MailUsageCC support@corp.com

Configuring Threshold E-mail

RealServer can send notification messages when a specified percentage of RealServer's bandwidth or connection limits is exceeded. The bandwidth limit is specified by the MaxBandwidth setting (see page 102). The connection limit is specified by the ClientConnections setting (see page 102).

Use the following settings to specify the percentage usage of bandwidth or connections that causes RealServer to send a threshold notification e-mail message:

Setting	Description	Restart Server After Changing
MailUsagePeriod	Specifies the time period over which MailUsageThreshold is calculated.	No
MailUsageThreshold	Specifies the server usage level that causes RealServer to send a usage e-mail message.	No

The notification e-mail lists the number of licensed streams, the threshold exceeded, and the time for which the threshold was exceeded. For example, if the bandwidth threshold is exceeded:

```
In last 89 hours, server usage exceeded 92% for a
total of 5342 seconds
Licensed Streams: 812
>--- BANDWIDTH THRESHOLD exceeded ---<
Maximum Bandwidth: 413
Bandwidth Used: 398
Maximum Audio Connections: 750
Number of Connections Used: 732
```

MailUsagePeriod

Specifies the time period over which **MailUsageThreshold** is calculated. At the end of each **MailUsagePeriod**, the server resets the statistics used to calculate **MailUsageThreshold**.

Used by	pnservice
Default value	24
Range of values	Positive Integers

Restart Server after change	No
-----------------------------	----

MailUsagePeriod <hours>

Example

MailUsagePeriod 168

Makes RealServer reset the **MailUsageThreshold** statistics every week.

MailUsageThreshold

Percentage of maximum bandwidth (if specified) or total license streams that must be in use before a usage threshold e-mail is sent to the addresses specified in **MailMessageUser** and **MailUsageCC**.

Used by	pnservice
Default value	80
Range of values	1 to 100
Restart Server after change	No

MailUsageThreshold <percent>

If **MaxBandwidth** is set to a value, **MailUsageThreshold** is a percentage of the **MaxBandwidth** value. If no maximum bandwidth is specified, **MailUsageThreshold** is a percentage of the **ClientConnections** value. If **ClientConnections** is not set, **MailUsageThreshold** is a percentage of RealServer's licensed streams.

Example

MailUsageThreshold 85

Limiting Server Bandwidth and the Number of Simultaneous Connections

RealServer lets you control how much of your network resources are dedicated to media by letting you specify how much bandwidth RealServer can use. You can also control how many clients can connect to RealServer at the same time.

The following settings limit server bandwidth and the number of simultaneous client connections to the server. A detailed description of each setting follows the table.

Setting	Description	Restart Server After Changing
ClientConnections	Specifies the maximum number of simultaneous client connections to RealServer.	No
MaxBandwidth	Specifies the maximum bandwidth that RealServer can use on particular network connection.	No

ClientConnections

Maximum number of simultaneous client connections.

Used by	pnservice
Default value	Licensed number of streams
Range of values	0, 1 - 32767
Restart Server after change	No

The maximum number of simultaneous media connections that RealServer can support is determined by the license you purchase unless you are using a remote license server. The **ClientConnections** parameter lets you set a lower limit than that specified by your license.

To specify a greater number of connections than the license on this computer allows, you need to get more streams from a remote license server.

To specify a value, enter the following line in the **server.cfg** file:

```
ClientConnections      <count>
```

The maximum number of connections cannot usefully exceed the maximum number that the bandwidth of your network connection supports. If **ClientConnections** is not specified, RealServer uses the number of streams specified by your license. If a **LicenseKey** is not specified or if your license key expires, RealServer allows a minimum of 2 streams.

A value of 0 specifies the maximum number of streams allowed by the license key on this computer.

MaxBandwidth

Maximum bandwidth (in Kbps) that RealServer can use of a particular network connection's total bandwidth capacity.

Used by	pnservice
Default value	0
Range of values	Positive Integers
Restart Server after change	No

The default value of 0 does not explicitly limit bandwidth. In this case, however, bandwidth is limited indirectly by the number of simultaneous streams that RealServer can support. The maximum number of simultaneous streams is determined by the lesser of **ClientConnections** or the licensed number of streams.

```
MaxBandwidth <number>
```

Example

```
MaxBandwidth 750
```

Restricts RealServer to using half of a T1 connection's capacity.

Security

Aside from using a firewall, there are two ways that you can control access to media files served by RealServer:

- You can put URLs for media files on restricted Web pages.
- You can use the **ConnectControlList** setting to specify which network domains can access RealServer.

ConnectControlList

Specifies the addresses from which clients are allowed to access RealServer.

If you purchased an intranet license for RealServer, you must specify a **ConnectControlList** to enable the users on your intranet to access your RealServer.

If you purchased an Internet license for RealServer, you can optionally use **ConnectControlList** to restrict access to your Server.

Used by	pnservice
Default value	(none)
Range of values	Valid IP addresses
Restart Server after change	No

```
ConnectControlList [{<address>, <net mask>}, ...]
```

Where:

<address> is the domain address of the computer allowed to access RealServer. All four octets of the address must be specified.

<net mask> is a mask that specifies the bits in the domain address that are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards. For example, an address of 121.23.101.0 with a net mask of 255.255.255.0 accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted. The net mask 255.255.255.255 accepts only the single IP address specified.

Note Servers with intranet licenses cannot specify a net mask of 0.0.0.0.

To allow any player to connect, do not include a **ConnectControlList** setting in your configuration file. To prevent any player from connecting, specify:

```
ConnectControlList [{0.0.0.0, 255.255.255.255}]
```

Example

```
ConnectControlList  
[ {100.61.0.0, 255.255.0.0},  
  {204.71.154.0, 255.255.255.0},  
  {204.71.155.202, 255.255.255.255} ]
```

SplitterControlList

List of splitter domain names that are allowed to access RealServer. To use this setting, you must purchase splitting as part of your RealServer.

If you purchased an intranet license for RealServer, you must specify a SplitterControlList to enable splitters on your intranet to access your RealServer.

Used by	pnservice
Default value	(none)
Range of values	Valid IP addresses
Restart Server after change	No

SplitterControlList [{<address>, <net mask>}, ...]

where:

address is the domain address of the splitter computer allowed to access RealServer.

net mask is a mask that specified the bits in the domain address that are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards. For example, an address of 121.23.101.0 with a net mask of 255.255.255.0 accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128 all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted. The net mask 255.255.255.255 accepts only the single IP address specified.

Note Servers with intranet licenses cannot specify a net mask of 0.0.0.0.

To allow any player to connect, do not include a **SplitterControlList** setting in your configuration file. To prevent any player from connecting, specify:

```
SplitterControlList [{0.0.0.0, 255.255.255.255}]
```

Example

```
SplitterControlList [{204.71.154.0, 255.255.255.0}]
```

System Manager Settings

You can specify the maximum number of System Manager sessions that can connect to RealServer at the same time. You can also specify a password that System Manager must use to connect to RealServer.

MonitorConnections

Maximum number of System Manager sessions that can connect to RealServer.

Used by	pnservice
Default value	4
Range of values	Whole number greater than or equal to zero
Restart Server after change	No

The System Manager is a Windows application used to monitor a running RealServer. The System Manager connects to RealServer over a TCP/IP connection. The maximum number of these connections should be restricted to the number of System Managers that you anticipate running.

```
MonitorConnections <count>
```

The maximum number of System Manager connections does not reduce the allowed number of media connections.

Example

MonitorConnections 6

MonitorPassword

Password that System Manager must use to connect to RealServer.

Used by	pnserver, rssm
Default value	(none)
Range of values	Alpha-numeric string without spaces
Restart Server after change	No

MonitorPassword <password>

Example

MonitorPassword SrvTest1

Configuring Web Servers to Work with RealServer

RealServer works with any Web server that supports configurable MIME types. Setting the correct MIME type makes the user's Web browser play the contents of a media file with a RealPlayer rather than download the contents of the file.

Your Web Server needs to define the following MIME types:

audio/x-pn-realaudio (files with a .ra, .rm or .ram file extension)
audio/x-pn-realaudio-plugin (files with a .rpm file extension)

The procedure for associating media files with these MIME types varies from one Web server to another. If the Web server is on the RealServer machine, this is done by installers on Internet Explorer and Netscape on Windows NT, and on Apache, Netscape and NCSA on UNIX. The following procedures tell how to add MIME types to some common brands of Web servers. If you have questions, or if your Web server is not listed here, please consult your Web server documentation or the online documentation at the Progressive Networks Web site:

<http://www.real.com>

CERN HTTPD (v.3.0) Server

1. Add the following lines to the **httpd.conf** file under the server's root directory:

```
AddType .ram audio/x-pn-realaudio      binary
AddType .rpm audio/x-pn-realaudio-plugin binary
AddType .ra audio/x-pn-realaudio        binary
AddType .rm audio/x-pn-realaudio        binary
```

2. Reinitialize the Web server.

EMWAC HTTPS (Windows NT Only)

1. In Control Panel, start the HTTP server applet.
2. Click the **New Mapping** button.
3. In the Extension edit box, enter the filename extension:
RAM
4. In the Mime Type edit box, enter the full MIME type:
audio/x-pn-realaudio
5. Click the **OK** button.
6. Repeat Steps 3 and 4, using the file extension:
RPM
and the MIME type:
audio/x-pn-realaudio-plugin
7. Reinitialize the Web server.

Mac HTTP and HTTPD4Mac Servers

1. Enter the following information into your configuration file in the format appropriate for your server:

```
Action: TEXT
File Suffix: .ram
File Type: *
MIME Type: audio/x-pn-realaudio
Creator: *
```

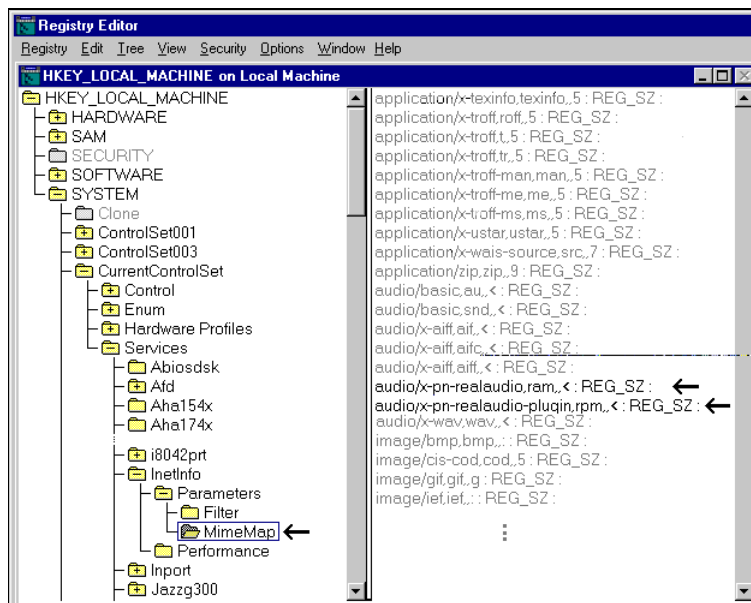
2. Repeat with the File Suffix:
.rpm
and the MIME Type:
audio/x-pn-realaudio-plugin

Microsoft Internet Information Server (Windows NT Only)

MIME type configuration is done in the Windows NT registry. To edit the registry:

1. Log on as Administrator.
2. Start **Regedt32**.
3. Click the entry:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
Services\InetInfo\Parameters\MimeMap



4. Click **Add Value** on the Edit menu.

5. In the **Add Value** box, enter:
`audio/x-pn-realaudio-plugin,rpm,,<`
6. In the **Data Type** box select:
`REG_SZ`
And click the **OK** button.
7. Leave the **String** box blank and click the **OK** button.
8. Repeat Steps 4 through 7. For Step 5, enter:
`audio/x-pn-realaudio,ram,,<`

NCSA HTTPD (v. 1.3 and 1.4) Server

1. In the file **srm.conf** in the **SERVER_ROOT/conf** subdirectory, add the following lines:

```
AddType audio/x-pn-realaudio .ram
AddType audio/x-pn-realaudio-plugin .rpm
AddType audio/x-pn-realaudio .ra
AddType audio/x-pn-realaudio .rm
```
2. Reinitialize the Web server.

Netscape Netsite Server

1. Add the following to the **MIME.types** file:

```
type=audio/x-pn-realaudio exts=ram
type=audio/x-pn-realaudio-plugin exts=rpm
```
2. Add the following line to the Server's main configuration file (called **magnus.conf** in the examples given in the Netsite documentation):

```
Init fn=load-types mime-types=mime.types
```
3. Reinitialize the Web server.

O'Reilly Website NT Server

Use the admin tool on the mapping page to change the content type by entering the following commands:

```
.ram audio/x-pn-realaudio
.rpm audio/x-pn-realaudio-plugin
.ra audio/x-pn-realaudio
.rm audio/x-pn-realaudio
```

Webstar and Webstar PS

1. Start the Admin program for the Webstar server.
2. On the Configure menu, click **Suffix Mapping**.
3. Enter the MIME type information into its associated fields exactly as shown in the following example (these fields are case sensitive):

```
Action: TEXT
File Suffix: .ram
File Type: *
MIME Type: audio/x-pn-realaudio
Creator: *
```

4. Click the **Add** button to update the MIME types directory.
5. Repeat Steps 3 and 4, using the File Suffix:

```
.rpm
```

and the MIME Type:

```
audio/x-pn-realaudio-plugin
```

Spinner 1.0b12 - 1.0b15 / Roxen 1.0

1. Point your browser to the following URL:

**http://<server_name>:18830/Configurations/Gnats/Contenttypes/
Extensions?40**

Where:

<server_name> is the name of computer running your Web server
<18830> is the default administration server port; change this port
number to your administration server port if necessary

2. Enter the MIME types into the dialog box.

Apache 1.1.1

Apache comes preconfigured, but the MIME type for RealAudio and RealVideo files needs to be changed from audio/x-realaudio to audio/x-pn-realaudio. MIME types are normally stored in **/usr/local/etc/httpd/conf**.

Configuring Advanced Server Settings

This chapter describes the settings in the server configuration file (**server.cfg**) that control more advanced server features. These features are not included in the basic RealServer license. Your server must be specially licensed to use these features.

The advanced configuration settings can be divided into the following groups:

- **Splitting settings**
- **Remote licensing settings**
- **ISP Hosting settings**

Splitting

RealServer provides support for UDP-based splitting. Splitting allows servers to transmit live data sent from other RealServers. The servers receiving data from another server are referred to as splitters. Splitting can increase flexibility and efficiency in delivering content to users. With RealServer, the server providing live content that it wants to split (send to other servers) can establish a connection port for the splitter to request the live streams. A splitter contacts the server on this port and establishes a UDP data stream between the two.

Configuring Splitting

In addition to offering recorded media files or live broadcasts sent directly from RealAudio and RealVideo Encoders, RealServer can now offer live media sent from another RealServer. This gives you greater efficiency and flexibility in delivering live broadcasts to users.



For example, say you want to broadcast a concert from Milan over the Internet. You can have RealServers in New York City and Tokyo receive the broadcast. Then users in those cities connect to the RealServer closest to them, thereby getting better media quality and performance.

The ability to split media streams is an optional feature controlled by the license you purchase from Progressive Networks. If you would like to add this capability to your network, contact your Progressive Networks reseller or Progressive Networks.

To split a media stream, you need:

- One RealServer supplying the media stream.
- Another RealServer licensed for splitting.

All streams from the live server are broadcast to the splitter. If two servers are sending to the same server and they have a stream with the same name, the first one will be the only one split.

The following configuration parameters are required for splitting:

```
SplitterControlList  
SplitterSourceList  
SplitterAnnouncePort
```

Setting the **SplitterBufferDelay** configuration parameter on the splitter is important for preventing dropouts in the media stream. The recommended value is 20 seconds; a minimum of at least 10 seconds should usually be used.

The following tables summarize the configuration settings that control splitting, and indicate whether it is necessary to restart the Server after changing the setting.

Required Splitter Setting

Setting	Description	Restart Server
SplitterSourceList	Specifies the Live Servers that this splitter should contact and split all live streams from.	No

Required Server Settings

Setting	Description	Restart Server
SplitterAnnouncePort	Defines the server port that a splitter will contact to request the splitting of it's live streams.	No
SplitterControlList	Defines the valid servers that can connect and obtain split streams from this server. Any servers that contact on the SplitterAnnouncePort and are not in this list are ignored.	No

The following settings should not be changed unless you need to tune or optimize the splitting system.

Splitter Tuning Settings

Setting	Description	Restart Server
SplitterBufferDelay [number of seconds]	Defines how much data that the Splitter will buffer before releasing the stream to players.	No
SplitterTimeout [number of seconds]	Defines the number of second the splitter will wait after a stream has been stopped or interrupted. Alter only if you wish to shutdown streams quicker or if you have streams stopping when they shouldn't.	No
SplitterSourceProbeInterval	Determines how often the splitter requests details on live streams from a server. This may require altering if the Timeout values are changed.	

Server Tuning Settings

Setting	Description	Restart Server
SplitterResendBuffer	Defines the depth of the buffer used for UDP resends in the Splitter protocol. Increase if you are getting consistent loss on all players from a split stream.	No

Setting	Description	Restart Server
SplitterSourceTimeout	Defines how long it takes the Server to stop sending data to a splitter when the splitter is not responding.	

Controlling Splitter Access to a Server

You can specify the splitters that are allowed to access a RealServer. If you do not limit the splitters, any splitter can access your server.

The **SplitterControlList** configuration parameter lists the addresses of splitters that are allowed to access your Server.

If you specify a **SplitterControlList** configuration parameter, only splitters from the specified addresses can access your Server. If you do not specify a **SplitterControlList** value, any splitter is accepted.

See “SplitterControlList” on page 121.

SplitterSourceList

Specifies the Live Servers that this splitter should contact and split all live streams from.

To use this setting, you must purchase splitting as part of your RealServer.

Used by	pnserver
Default value	none
Range of values	Any number of pairs of any valid IP address or valid domain name and any valid port
Restart Server after change	no

```
SplitterSourceList <[{host, port}]
```

Example

```
SplitterSourceList <[{172.16.2.54,5780}  
{norton.prognet.com, 5781}]>
```

where:

host is any valid IP address or valid domain name.

port is any valid port.

SplitterAnnouncePort

Defines the server port that a splitter will contact to request the splitting of it's live streams.

To use this setting, you must purchase splitting as part of your RealServer.

Used by	pnserver
Default value	0
Range of values	Valid UDP port
Restart Server after change	no

```
SplitterAnnouncePort <port#>
```

Example

```
SplitterAnnouncePort <8090>
```

where:

port is any valid UDP port.

SplitterControlList

List of splitter domain names that are allowed to access RealServer. To use this setting, you must purchase splitting as part of your RealServer license.

Used by	pnservice
Default value	(none)
Range of values	Valid IP addresses
Restart Server after change	No

SplitterControlList [{<address>, <net mask>}, ...]

where:

address is the domain address of the splitter computer allowed to access RealServer.

net mask is a mask that specifies the bits in the domain address that are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards. For example, an address of 121.23.101.0 with a net mask of 255.255.255.0 accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted. The net mask 255.255.255.255 accepts only the single IP address specified.

To allow any player to connect, do not include a **SplitterControlList** setting in your configuration file. To prevent any player from connecting, specify:

SplitterControlList [{0.0.0.0, 255.255.255.255}]

Example

SplitterControlList [{204.71.154.0, 255.255.255.0}]

SplitterBufferDelay

Amount of media, in seconds, to store in the TCP buffer for splitting. Buffering helps reduce packets loses (dropouts) over a splitter connection.

The recommended value is 20 seconds; a minimum of at least 10 seconds should usually be used. This parameter is set on the RealServer acting as a splitter for another Server.

To use this setting, you must purchase splitting as part of your RealServer license.

Used by	pnservice
Default value	0
Range of values	positive Integers
Restart Server after change	no

```
SplitterBufferDelay <time>
```

Example

```
SplitterBufferDelay 20
```

where:

time is the amount of media, in seconds, to be stored in the TCP buffer for splitting.

SplitterTimeout

Defines the number of seconds the splitter will wait after a stream has been stopped or interrupted. Alter only if you wish to shutdown streams quicker or if you have streams stopping when they shouldn't.

To use this setting, you must purchase splitting as part of your RealServer.

Used by	pnserver
Default value	20
Range of values	0-32767
Restart Server after change	no

`SplitterTimeout <timeout>`

Example

`SplitterTimeout <40>`

where:

`timeout` is the number of seconds the splitter will wait after a stream has been stopped or interrupted.

SplitterSourceProbeInterval

Determines how often the splitter requests details on live streams from a server. This may require altering if the Timeout values are changed.

To use this setting, you must purchase splitting as part of your RealServer.

Used by	pnserver
Default value	10
Range of values	0-32767
Restart Server after change	no

```
LiveSourceProbeInterval <source probe interval>
```

Example

```
LiveSourceProbeInterval <20>
```

where:

`source probe interval` is how often the splitter requests details on live streams from a server.

SplitterResendBuffer

Defines the depth of the buffer used for UDP resends in the Splitter protocol. Increase if you are getting consistent loss on all players from a split stream.

To use this setting, you must purchase splitting as part of your RealServer.

Used by	pnservice
Default value	20
Range of values	0-32767
Restart Server after change	no

```
SplitterResendBuffer <resend buffer>
```

Example

```
SplitterResendBuffer <30>
```

where:

`resend buffer` is the depth of the buffer used for UDP resends in the Splitter protocol.

SplitterSourceTimeout

Defines how long it takes the Server to stop sending data to a splitter when the splitter is not responding.

To use this setting, you must purchase splitting as part of your RealServer.

Used by	pnservice
Default value	40
Range of values	0-32767
Restart Server after change	no

`SplitterSourceTimeout <source timeout>`

Example

`SplitterSourceTimeout <60>`

where:

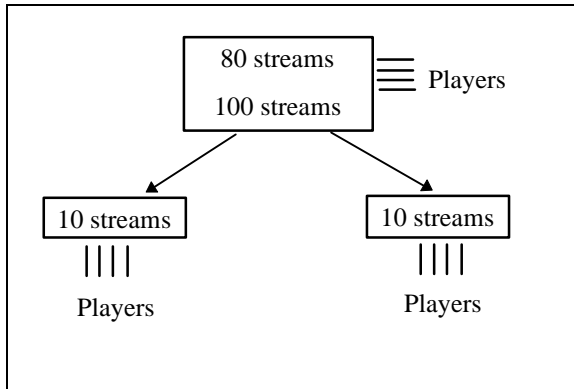
`source timeout` is how long it takes the Server to stop sending data to a splitter when the splitter is not responding.

Configuring Remote Licensing

You can share one RealServer’s allotted number of streams among several other RealServers. This simplifies license management when you own several RealServers.

The ability to share allotted streams is an optional feature controlled by the license you purchase from Progressive Networks. If you would like to add this capability to your network, contact your Progressive Networks reseller or Progressive Networks.

For example:



To share a stream license you need:

- One RealServer licensed and configured to share allotted streams. This RealServer is called the remote license server.
- Other RealServers configured to accept stream allotment. These RealServers are called the license clients.

Remote License Server Configuration

You need to specify the names of the license clients and the number of streams allotted to each. You specify this information with the LicenseClients configuration setting:

```
LicenseClients [ {<host>:<port>,<streams>} , ... ]
```

where:

`host` is the name of a license client.

`port` is the PnaPort value of the license client.

`streams` is the number of streams to allocate to the license client.

License Client Configuration

For a license client to receive stream allotment from a remote license server, you need to specify the name of the remote license server and the port number it uses to communicate with the license client. You also need to specify the

number of streams that can be allocated to the client. Add the following entries to the configuration file of each license client:

```
RemoteLicenseHost    <host>
RemoteLicensePort    <port>
ClientConnections    <count>
```

where:

`host` is the name of the remote license server.

`port` is the number of the port the remote license server uses to connect to the license client, specified with the **LicenseClients** configuration setting on the remote license server.

`count` is the total number of licenses available to this client. This value should be the sum of the `streams` value on the `LicenseClients` parameter of the remote license server and the number of locally licensed streams.

Note The **RemoteLicenseHost** and **RemoteLicensePort** settings must specify a different RealServer from the one on which this configuration file is used.

LicenseClients

List of servers that can request license information from a remote license host. To use this setting, you must purchase remote license management as part of your RealServer license.

Used by	pnservice
Default value	(none)
Range of values	N/A
Restart Server after change	No

```
LicenseClients [ {<host>:<port>,<streams>} , ...]
```

where:

host is the name of a license client.

port is PnaPort value of the license client.

streams is the number of streams to allot to the license client.

Example

```
LicenseClients  
[ {chris.inet.com:7075,50}, {yuri.f4.com:7073,75} ]
```

RemoteLicenseHost

Name of the RealServer acting as a remote license host. To use this setting, you must purchase remote license management as part of your RealServer license.

Used by	pnserver
Default value	(none)
Range of values	Valid host name
Restart Server after change	Yes

```
RemoteLicenseHost <name>
```

Note The **RemoteLicenseHost** setting must specify a different RealServer from the one on which this configuration file is used.

Example

```
RemoteLicenseHost license.prognet.com
```


RemoteLicensePort

The TCP port used by the license client to connect to the remote license host. To use this setting, you must purchase remote license management as part of your RealServer license.

Used by	pnservice
Default value	7071
Range of values	Valid port number
Restart Server after change	Yes

```
RemoteLicensePort <port>
```

Example

```
RemoteLicensePort 7075
```

Note The **RemoteLicensePort** setting must specify a different RealServer from the one on which this configuration file is used.

Configuring for ISP Hosting

Creating Accounts on RealServer

With RealServer you can divide the stream capacity of your server between multiple accounts. For example, this can let an Internet Service Provider (ISP) buy RealServer and then allocate media streams to individuals, companies, or organizations who use the ISP to host their Web sites on the Internet.

The ability to allocate is an optional feature controlled by the license you purchase from Progressive Networks.

You can create individual accounts and specify the number of streams allocated to each, or you can use a naming convention to allocate the same number of streams to a large number of accounts.

UserList entries can not be added or deleted from the System Manager. The **UserList** entry only supports changes to existing entries from the System Manager. For example, you can change the maximum or minimum number of connections a particular account is authorized. To make more substantial changes to the **UserList** you can edit **server.cfg** using a text editor.

Creating Individual Accounts

To divide your media stream capacity between specified individual accounts, you need to specify which users, the location of the user's files, and the minimum and maximum number of streams they are guaranteed. Add the following entry to your configuration file:

```
UserList [ {<user>, <path>, <min>, <max>}, ...]
```

where:

user is the name of the user. This defines the key that the URL passes to RealServer to allow selection of a particular account entry. Name does not have to be a user directory and can be a string up to 1024 characters. In the URL, name is preceded by a tilde (~). For example:

```
pnm://audio.real.com/~fred/test.ra
```

Selects the account entry defined for the user **fred** and then plays the media file **test.ra** from the **fred** privateRPath directory.

path is the path of the directory of user files. This configuration setting creates a separate path for media files in each account. This lets the owner of the account alter the files in their own directory without granting them access to any other user's files.

min is the minimum number of streams allocated to the account. These streams are no longer available to any general RealServer requests. If **min** is 0, no streams are reserved for that account.

max is the maximum number of streams allocated to the account. This number can be from 0 to the total number of streams available on RealServer.

For example, to allocate eight streams between two businesses posting Web sites on your service, use:

```
UserList [ {ElectroMotors, /usr/electro/ra, 2, 5},  
           {CityWeld, /usr/cityweld/ra, 1, 3} ]
```

If more than the available streams are allocated to individual account entries, RealServer logs an error and provides access to the minimum streams for those entries in **UserList** before the limit is exceeded. All account entries after the limit is exceeded are not allocated streams.

Creating Accounts Using a Naming Convention

If you need to create a large number of accounts, and allocate the same number of streams to each, you can use a naming convention instead of listing each account individually. This function is typically used by Internet Service Providers who make RealServer available to a large number of customers.

The ability to create accounts using a naming convention is an optional feature controlled by the license you purchase from Progressive Networks. If you would like to add this capability to your Server, contact your Progressive Networks reseller or Progressive Networks.

You can define a naming convention for most accounts, and still create individual accounts with different numbers of streams.

You can use one or both of the following naming conventions to allocate large numbers of accounts.

Naming Convention One

All accounts using this naming convention have a URL with the following format:

```
pnm://server.com/~account/directory/file.ra
```

All URL requests that begin with the same value for `account` are counted against that account's stream allocation.

The files for this account must be located in the **/directory/** directory relative to the path specified in the following UserList entry.

The following special UserList entry specifies the number of streams allocated to each account that uses this naming convention:

```
{ ~*, <path>, <min>, <max> }
```

In either of these cases, on UNIX you can use the special character `~` as the `<path>` entry and the server will use the account to look up the password entry for this user and use their home directory to locate the content.

Naming Convention Two

All accounts using this naming convention have a URL with the following format:

```
pnm://server.com/dir1/dir2/dir3/dir4/file.ra
```

All URL requests that begin with the same value for the specified number of directory levels are counted against that account's stream allocation. If the directory level is set to 3, then `/dir1/dir2/dir3/` becomes the unique account identifier.

The files for this account must be located in the **/dir1/dir2/dir3** directory relative to the path specified in the following UserList entry.

The following special UserList entry specifies the number of streams allocated to each account that uses this naming convention:

```
{ *n, <path>, <min>, <max> }
```

Where `n` is the number of directory levels that make up the unique account.

Note If you use the `*n` naming convention, you must use it for all files on that RealServer. URLs relative to the Server BasePath do not work.

UserDir

Path to be appended to the path defined for the account entries defined in the **UserList** setting. To use this setting, you must have hosting as part of your RealServer license.

Used by	pnserver
Default value	(none)
Range of values	Valid path name
Restart Server after change	No

UserDir <string>

If no **UserDir** is specified then RealServer looks for media files in the path specified in the **UserList** entry.

Example

UserDir content

makes RealServer look for media files in the subdirectory **content** of the directory specified in the **UserList** entry.

UserList

List of accounts that are allocated media streams for private use. To use this setting, you must have hosting as part of your RealServer license.

Used by	pnserver
Default value	(none)
Range of values	N/A
Restart Server after change	No

This field is used to create RealServer Hosting.

```
UserList
    [ {<Accnt>, <privateRApath>, <minStreams>,
      <maxStreams>}, ... ]
```

There can be as many entries in this list as required. For more information on hosting, see “Creating Accounts on RealServer” on page 129.

Two special values (~* and *n) for <Accnt> enable you to define accounts using naming conventions. You can have a large number of accounts without having to list them individually. For more information, see “Creating Accounts Using a Naming Convntion” on page 131.

Example

```
UserList
[ {~*, /usr/persacct/, 1, 2},
  {ElectroMotors, /usr/electro/ra, 2, 5},
  {CityWeld, /usr/cityweld/ra, 0, 3}]
```

Configuring for Intranets

ConnectControlList

List of domain names that are allowed to access RealServer.

If you purchased an intranet license for RealServer, you must specify a **ConnectControlList** to enable the users on your intranet to access your RealServer.

If you purchased an Internet license for RealServer, you can optionally use **ConnectControlList** to restrict access to your Server.

Used by	pnserv
Default value	(none)
Range of values	Valid IP addresses

Restart Server after change	No
-----------------------------	----

```
ConnectControlList [{<address>, <net mask>}, ...]
```

Where:

address is the domain address of the computer allowed to access RealServer. All four octets of the address must be specified.

net mask is a mask that specifies the bits in the domain address that are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards. For example, an address of 121.23.101.0 with a net mask of 255.255.255.0 accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted. The net mask 255.255.255.255 accepts only the single IP address specified.

Note Servers with intranet licenses cannot specify a net mask of 0.0.0.0.

To allow any player to connect, do not include a **ConnectControlList** setting in your configuration file. To prevent any player from connecting, specify:

```
ConnectControlList [{0.0.0.0, 255.255.255.255}]
```

Example

```
ConnectControlList  
[{100.61.0.0, 255.255.0.0},  
{204.71.154.0, 255.255.255.0},  
{204.71.155.202, 255.255.255.255}]
```

Problem Solving

Using the Access and Error Log Files

The **pnerror.log** and **access.log** files reside in the **logs** subdirectory of your RealServer installation. The error log records information and error messages about RealServer operation. The access log records transactions by clients.

Reading Log Files

The error and access log files are stored as plain text. You can read them using a text editor or word processor.

You should read your log files on a regular basis. How frequently you read them depends on the amount of traffic your RealServer handles and if you are encountering any problems.

For information on the structure and content of the log files, see “Access and Error Log Messages” on page 138.

Changing Log Files

Because new information is appended to log files for each error and transaction, log files can grow quickly. To keep your log files at a manageable size, you should change them on a regular basis. You may want to archive log files to maintain a record of your server’s performance.

Windows NT

Changing the log files on Windows NT requires changing the name of the log file set in the configuration file.

1. Connect System Manager to the RealServer with the log file you want to change.
2. On the Server menu, click **Configuration**.
3. Enter the new name for the log file and click the **OK** button.

ErrorLogPath for the error log

LogPath for the access log

RealServer starts writing to the new file.

For complete information on System Manager, see “Monitoring Performance” on page 160.

UNIX

There are several ways to change log files:

- You can use System Manager to change the log files. See the instructions in the previous section for Windows NT.
- You can rename the log file and UNIX continues to write to the renamed file until you enter a SIGHUP signal. RealServer. Then closes the existing, now renamed log file. When the next message needs to be logged, RealServer opens the log file using the settings in the configuration file.

For example, to change your access log file named **pnaccess.log**, rename it **access1.log**. RealServer continues to write to **access1.log**. Once RealServer receives a SIGHUP, it closes **access1.log** and opens **pnaccess.log** with the next message to be logged.

- If you do not want to keep your log files, simply delete the desired log file and issue a SIGHUP signal. Once RealServer receives the signal, it opens a new file. To send a SIGHUP signal, use the **kill** command with the server's process id. If you are in the logs directory, use the following syntax:

```
kill -hup `cat pnservice.pid`
```

Access and Error Log Messages

RealServer writes important status information to the access and server log files. This chapter describes the structure and contents of these log files. For information about reading and maintaining log files, see “Using the Access and Error Log Files” on page 136.

RealServer Access Log

The access log helps you monitor and manage your RealServer. You can view how many clients have connected to your server, the name of the client machines, the clips they listened to, the times of day they connected, and errors that were generated by RealServer. This information can give you an idea of who your audience is and what clips are popular.

The RealServer Log records transactions in the file format common to most Web servers. Each transaction is recorded on one line in fields delimited by spaces. To view the RealServer log, open the file specified by the name used in your LogFilePath using a word processor or text editor.

Two configuration parameters control what is written to the access log: **LoggingStyle** and **StatsMask**. Each field description lists any parameter settings required for that field to be included in the access log.

The access log format is:

```
<IP_address> <- -> <timestamp> "<GET filename>  
<protocol>" <return_code> <bytes_sent>  
[<client_ID_string>] [<client_unique_id>] <stat1>  
<stat2> <file_size> <file_time> <sent_time> <resends>  
<failed_resends>
```

Where:

<IP_address> IP address of Client.

For example:

123.45.678.90

<- -> Two hyphens for compatibility with Web server log formats.

<timestamp> Time that Client accessed the file in the format:

[<day>/<month>/<year>:<hh>:<mm>:<ss> <TZ>]

where TZ is the time zone expressed as the number of hours relative to the Coordinated Universal Time (Greenwich, England).

For example:

[31/Oct/1996:13:44:32 -0800]

<GET filename> File requested by Client. Filename includes the path relative to the Server's **BasePath** value.

For example:

GET /bands/fourfrosh/classics.ra

<protocol> Protocol and version used by Client in the format:

PNA<type>/<number>

where:

<type> is **T** for TCP connections, **H** for PNAviaHTTP, **M** for Multicast or blank for UDP connections. Type appears only if the **LoggingStyle** configuration parameter is set to 1.

<number> is the PN protocol number.

For example:

PNA/8
PNAT/8

<return_code> Return code using HTTP standard error codes. Always 200, meaning successful transfer.

<bytes_sent> Number of bytes transferred to Client during playback. This field can be lower than the total size of the media file, indicating partial

playback of the file. If this field is consistently low for some or all media files, this can mean that RealPlayers are able to connect to your server, but are unable to play files. Check your system error logs for messages relating to network system errors.

[<client_ID_string>] Client ID string. This field is not part of the common Web server access log format. The ID string is text sent by the Client that describes the version and type of Player being used. RealAudio Player versions 2 and 3 use the following underscore delimited format:

```
<platform>_<version>_<player>_<type>_<dist>_  
<language>_<CPU>
```

Where:

<platform> is the operating system that RealPlayer is running on—Win16, WinNT, Mac, and so on.

<version> is the operating system version number.

<player> is the version number of RealPlayer.

<type> is the type of RealPlayer.

<dist> is the distribution code of RealPlayer.

<language> is the code of RealPlayer. EN is US English.

<CPU> is the type of processor running the platform. If the processor does not have a hardware Floating Point Unit, the string “no-FPU” is appended to the end of the CPU field without a delimiter.

For example:

```
Win95_4.0_3.0.0.19_play32_PN01_EN_586
```

Note 1 RealAudio Player version 1 uses a different ID string in the following format:

```
<platform><player>
```

The field descriptions are the same as the newer format. For example:

```
Win1.0.0
```

Note 2 If the client is a splitter, the Client ID field contains the following string:

splitter

<client-unique_id> Unique ID generated during RealPlayer installation. It allows you to track details for individual players, enhancing the quality of the captured statistics. This entry is only displayed when LoggingStyle is set to 2.

Note This will also display the new entries for LoggingStyle 1.

<stat1> Connection statistics sent by the Client when it completes playing a clip. These optional fields are sent only when the **StatsMask** configuration parameter is set to 1 or 3. The Player user can also set a preference value to block sending connection statistics. When the Client blocks connection statistics, when the Client is a splitter, or when **StatsMask** is set to 0, the <stat1> and <stat2> fields are replaced by [UNKNOWN].

The connection statistics field starts with the string “Stat1” and has the following format:

```
[Stat1: <total> <order> <missing> <early> <late>
<format>]
```

Where:

<total> is the total number of packets received by the Client.

<order> is the number packets received by the Client out of order. These packets are reordered as they are being played by the Client.

<missing> is the number of missing packets that the Client did not receive. This is the most common problem reported on the PN Server Log. A low percentage of missing packets does not have a serious effect on quality; a high percent seriously degrades media quality. For more information, see “Network Performance Considerations” in the Server Operations section of this manual.

<early> is the number of packets received too early by the Client. If the Client receives any packets too early, then older packets are discarded. This problem is very rare, and it may indicate that the client’s machine is running too slow, or has a bad Internet connection. However, if this problem shows up very often, you need to investigate further.

<late> is the number of packets received too late by the Client. If the Client receives packets too late, the Player will have already played that portion of the media. Normally, this is a very rare occurrence; if it happens often, your Server's Internet connection may not be fast enough.

<audioformat> is the name of the decoder used to play the clip.

Values are:

dnet	RealAudio 3.0 formats
28.8	RealAudio 2.0 28.8 format
lpcJ	RealAudio 2.0 14.4 format

For example:

```
[Stat1: 641 0 0 0 0 dnet]
```

<stat2> Extended connection statistics sent by the Player when it completes playing a clip. These statistics are supported by all Players.

These optional fields are sent only when the **StatsMask** configuration parameter is set to 2 or 3. The Player user can also set a preference value to block sending connection statistics. When the Client blocks connection statistics, when the Client is a splitter, or when **StatsMask** is set to 0, the <stat1> and <stat2> fields are replaced by [UNKNOWN].

The extended connection statistics field starts with the string "Stat2" and has the following format:

```
[Stat2: <bandwidth> <available> <highest>  
<lowest> <average> <requested> <received>  
<late> <rebuffering> <type> <startup>  
<videoformat>]
```

Where:

<bandwidth> is the bandwidth in bits per second of the clip.

<available> is the average bandwidth in bits per second available to the user while the clip was playing.

<highest> is the highest time in milliseconds between the Client requesting a resent packet and receiving the packet.

<lowest> is the lowest time in milliseconds between the Client requesting a resent packet and receiving the packet.

<average> is the average time in milliseconds between the Client requesting a resent packet and receiving the packet for all resent packets.

<requested> is the number of resent packets requested by the Client.

<received> is the total number of resent packets received by the Client.

<late> is the number of resent packets received by the Client too late.

<rebuffering> is the rebuffering percentage for the clip.

<type> is the transport type for the connection. Values are:

0	UDP
1	TCP
2	IP Multicast
3	PNAviaHTTP

<startup> is the time in milliseconds from the Client sending the first packet to the Server to the Client receiving the first packet from the Server.

<video format> is the name of the video decoder used to play the clip. Values are:

pnrv - RealVideo 1.0
clrv - RealVideo (Fractal)

For example:

```
[Stat2: 15234 15552 0 0 0 0 0 0 0 0 0 220 28.8]
```

<file_size> Total amount in bytes of media data in the media file. This number is less than the size of the media file because it does not include the file header and other non-media information stored in the file. This field appears only if the **LoggingStyle** configuration parameter is set to 1.

Note This field is always 0 for live broadcasts.

<file_time> Total length, in seconds, of media stored in the media file. This field appears only if the **LoggingStyle** configuration parameter is set to 1.

Note This field is always 0 for live broadcasts.

<sent_time> Total length, in seconds, of the media sent to the Player. This field appears only if the **LoggingStyle** configuration parameter is set to 1.

<resends> Number of packets successfully resent because of transmission errors. This field appears only if the **LoggingStyle** configuration parameter is set to 1.

<failed_resends> Number of packets not successfully resent in time to correct transmission errors. This field appears only if the **LoggingStyle** configuration parameter is set to 1.

The following example shows three access log entries:

```
172.16.2.139 - - [04/Nov/1996:14:45:57 -0700] "GET
newclips/realcool.ra PNA/8" 200 590976
[Win95_4.0_3.0.0.19_play32_PN01_EN_586] [Stat1:
2592 0 0 0 0 28.8][Stat2: 15234 15552 0 0 0 0 0 0 0
0 220 28.8] 590976 310 310 0 0

172.16.2.139 - - [04/Nov/1996:14:53:49 -0700] "GET
classic/xyz144.ra PNAT/8" 200 4
[Win95_4.0_3.0.0.19_play32_PN01_EN_586] [UNKNOWN]
5580 5 0 0 0

172.16.2.139 - - [04/Nov/1996:16:01:10 -0700] "GET
speeches/carter.ra PNA/5" 200 55680 [Win1.0.0]
[Stat1: 229 0 0 0 0] 630020 630 55 0 0
```

RealServer Error Log

The error log helps you monitor and manage your RealServer. It contains both information and error messages about server operation. By looking for patterns of errors, you can troubleshoot and correct possible problems on your site.

To view the RealServer log, open the file specified by the name used in your **ErrorLogFilePath** using a word processor or text editor.

Error messages are recorded in the error log in the following format

```
[Date] [Time] [Servername](ProcessID) : [Error
Message]
```

A sample error message looks like this:

```
***15-Nov-96 14:13:30.488 myserver(1556): 6220:
No such user: joe
```


Note You can also have RealServer send messages to your e-mail address to notify you when certain thresholds are exceeded. See “Threshold Notification E-Mail” on page 99.

Common Error Messages

The following is a list of the more common error messages you might encounter:

Could not allocate enough file descriptors to meet capacity. Capacity has been set to <connection number>

The number of simultaneous connections has exceeded the capacity of your operating system. RealServer has automatically reset the number of media connections allowed to connect.

Invalid license key or information

Either you have not specified any licensing information for the LicenseKey setting in server.cfg or the licensing information you entered was incorrect. Check to make sure the information was entered exactly as you received it.

This license is for another platform

The license information you specified for the LicenseKey setting in server.cfg is for a different operating system. Check to make sure that you installed RealServer on the proper machine.

Server cannot be started before <date>

The RealServer license you purchased does not become valid until the date listed. Because RealServer requires a valid license to operate, your RealServer will not start until the date listed.

Server cannot be started after <date>

The RealServer license you purchased is not valid after the date listed. RealServer requires a valid license to operate.

Your license does not support ISP Hosting.

Your configuration file contains Hosting Service settings, but your license does not include Hosting Service. The Hosting Service configuration settings are ignored.

You must restart the server for this change to take effect.

You have made a change in `server.cfg` that will not take effect until you restart RealServer.

Out of Memory

RealServer is unable to dynamically allocate enough memory to create a new connection or manage existing connections. If you receive an Out of Memory message, you may require additional memory or you may need to add swap space for your RealServer machine to use for dynamic memory allocation.

Error retrieving <file name>

A user tried to access a file and the file could not be found. The user may have supplied the wrong URL and the Server rejected the request. However, if you see this more than once for the same file, you should check your metafile to ensure that the URL pointing to the file is accurate.

SIGPIPE Received, code:13 (UNIX systems only)

The SIGPIPE signal is sent to RealServer by the operating system when the client abruptly drops the connection. No action is required for this message.

Error retrieving URL <file name> (Codec error)

Error retrieving URL <file name> (Insufficient bandwidth)

The Player requested a file for which it does not have the correct CODEC installed or for which it does not have sufficient bandwidth to play. For example, a RealAudio Player 2.0 requesting RealAudio 3.0 content generates this error message.

Server General

General Server messages you might encounter include:

Cannot open ACCESS for logging

Can not get resource limit: <oserrormessage>

Can not set resource limit: <oserrormessage>

```
gethostname failed <errorno>
OS limit exceeded; max connections set to
<connectionnumber>
Illegal URL <url>
Invalid URL: <url>
Invalid bandwidth request: <bandwidthpath>
Invalid bandwidth request: <bandwidthpath>
SIGHUP received, code: 1
SIGINT or SIGTERM received, code: 2
<processid> exited
Terminating with exit code %d
Event file is corrupt
RTTPl Monitors not allowed
New live connection dropped due to server limit
New client connection dropped due to server limit
New monitor connection dropped due to server limit
New subserver connection dropped due to server limit
```

Server Communication

Server communication messages you might encounter include:

```
<connectionid>: Illegal hello message: <data>
<connectionid>: Version %d protocol not supported
```

Licensing

Server licensing messages you might encounter include:

```
Server expired, no new connection will be accepted
Invalid license key or information
This license is for another platform
Server cannot be started before <date>
Server cannot be started after <date>
Your license does not permit ISP Hosting
```

Server Configuration

Server configuration messages you might encounter include:

Expected ',', ']' or ']' in list at line <lineno>
Invalid number of elements in struct at line <lineno>
Invalid punctuation token '<character>' in config file
Invalid integer at line <lineno>
Expected type to be a string or an int
Expected type to be a string
Expected type to be an int
Invalid integer
Negative values not allowed
Expected type to be a list
Expected type to be a struct
Invalid configuration
You must restart the server for this change to take effect
Could not verify that BasePath is valid
You must specify a base path. If you want to specify the current working directory, then use '.'
Invalid group id
Invalid group name
Can't change to group id
Invalid user name
Invalid user id
Can't change to user id
Invalid license
This license is not yet valid
This license has expired
Invalid error log path
Invalid log path
You have allowed more user max streams than the server can support
You have reserved more user min streams than the server can support

Can't open the pna port
Invalid platform in the license
Invalid timeout
MaxThreads must be greater than zero
ClientConnections must be greater than MaxThreads
Min streams for user '<username>' exceeds max streams
User '<username>' already in UserList, skipping
User max streams exceed licensed client connections
User min streams exceed licensed client connections
Can't open config file server.cfg
MaximumClientConnections too large,
No such group: <groupname>
Must specify a number for Group config variable
No such user: <username>
Must specify a number for User config variable

Server Technical

Server technical messages you might encounter include:

Bad header: ileave: <no> gran: <no>, channels: <no>,
frame: <no>, bpm: <no>
Forked subserver count exceeded
No read streams for select
<connectionid>: bad player
<connectionid>: read: <oserrormessage>
<connectionid>: write: <oserrormessage>
Unsupported event type 0x%x
<connectionid>: unknown error state <errorstate>
<connectionid>: port <portno>: <oserrormessage>
<connectionid>: monitor rejected
Invalid opcode: <opcodeno>
Accept on port <portno>: <oserrormessage>
SIGCHLD received, pid <processid> status <exitcode>
SIGPIPE received, code: 13

```
Socket initialization failed with error <errorcode>
WSACleanup failed %d
Bad magic string for event file
Version %d is incorrect for event file
Unknown entity type: %d
get_conn: error: <oserrormessage>
<connectionid>: Client broke connection in start state
<connectionid>: Client broke connection in key state
```

Troubleshooting RealServer

If you are experiencing problems with RealServer, you need to use RealPlayer to test links on your site to isolate the source of the problem. Before you try to connect to your site, launch System Manager to see if your RealServer has an available connection for you to use. If your RealServer has a license that includes Hosting Service, you can use hosting to reserve a stream for your own testing.

The access and error logs record errors and information generated by RealServer. For instructions about how to open and interpret the log file, see "Using the Access and Error Log Files" on page 136.

The following questions can help diagnose the problem:

Is RealServer running on the host machine?

Use `ps` (on UNIX), or the Services Control Panel (on Windows NT) to check if RealServer is running. If the Server is not running, start the server.

Is the IP address of the host machine correctly configured in the network routers?

If the Player cannot access the Server over the network, then you cannot expect media to play. Configuring IP address and routers is a complex issue. Contact a networking specialist for help.

Is the machine you are using to test the media connected to the network used by the Server host computer?

You must have a network connection between Player and Server for media to play. Contact a networking specialist for help.

Is there a firewall between the Player and the Server?

You need to configure your system's firewalls to permit media to play through them; for details see "Security" on page 104.

Can you connect to the RealServer with the System Manager?

The System Manager application can help you diagnose the problem by validating communications between the Player and RealServer and by letting you view the running state of the connection during attempts to play media.

Is the media file downloading to the Player instead of playing in real time?

Media files cannot be referenced directly by your Web document. Remember that the Web page is being served by a Web server, but the media file is being served by RealServer. The Web page must point the user's Web browser to the media file by way of a metafile, which is a text file you create and save with a .ram extension. The metafile contains the URL of the .ra file located on your RealServer. The Web page contains a link to the metafile. For information on metafiles, refer to "Metafiles" on page 327.

Is there unreadable text displaying on the screen instead of media?

You have not configured your Web server to recognize RealAudio and/or RealVideo MIME types. For information on RealServer and MIME types, see "Configuring Web Servers to Work with RealServer" on page 109.

If you still have problems after considering these possibilities, please contact Progressive Networks at:

<http://www.real.com>

Server Operation

This chapter tells you how to start and stop RealServer, read log files to help diagnose problems, and fix some common problems.

Starting RealServer

Starting RealServer on a Windows NT Computer

Starting RealServer Manually

To start RealServer manually:

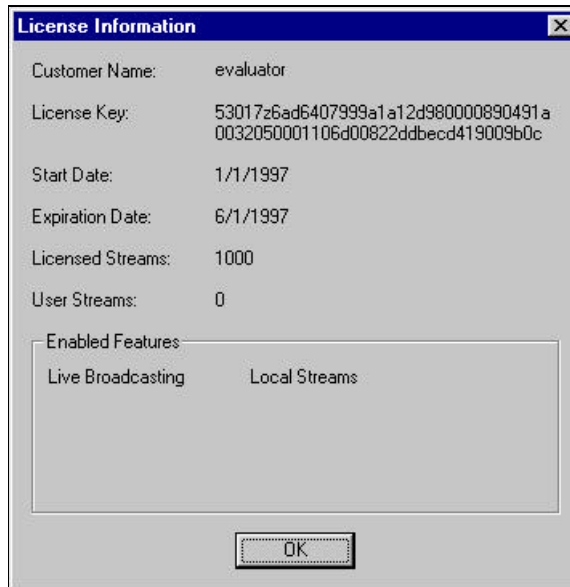
1. From Start, select **Programs**.

Note You can also double-click on the RealServer Control Center icon on your desktop to open.

2. From Real, select **RealServer Control Center** (or you can execute the **svrctrl.exe** application in the \bin directory. The RealServer Control Center dialog box displays. The status bar displays the status of the Server. It can be running or stopped.

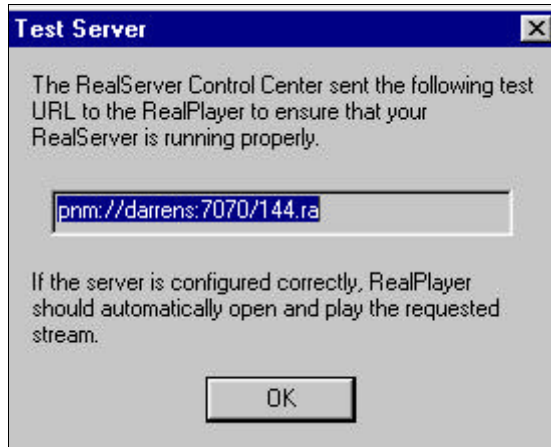


3. The default configuration file, `server.cfg`, is automatically loaded. To select a different configuration file, from **File**, select **Open**. The Open dialog box displays.
4. Choose a configuration file. Click **Open**. The license key and name of the file are checked. If either is invalid, a warning will display.
 - If you receive a warning, from **View**, select **License**. The details or your license is displayed. This information is helpful should you need to speak with technical support.

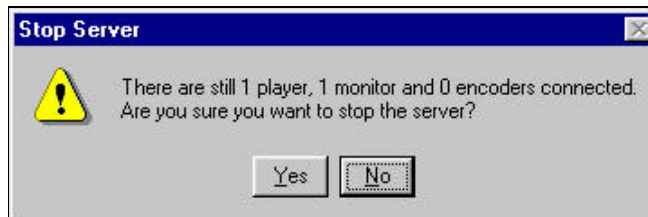


5. From **Server**, select **Start**. The Server starts. The status bar displays the status of the Server.

6. If RealPlayer is installed on the current machine, you can test the Server while it is running. From Server, select **Test**. RealPlayer opens, locates and displays an appropriate file. (If RealPlayer is not installed, a warning will display.)



7. On the menu bar, select **Sites**. RealServer Control Center will automatically direct your browser to either RealHome Page or Timecast.
8. On the menu bar, select **Help**. Notice the links to pages on our Web site that provide help.
9. If you want to stop the server without quitting the application, select Server then click **Stop**.
10. From File, select **Exit**. Notice a confirmation dialog box displays, telling you that exiting Server Control Center will stop the Server. If users are connected to the Server, you will be notified.



To start RealServer manually from the command line:

1. Change to the directory where you installed RealServer.
2. Start RealServer by entering:

```
bin\pnserver server.cfg
```

RealServer does not return any messages to indicate that it has started, and there is no prompt on the screen for as long as it is running.

If RealServer does not start, review the error messages in the RealServer Log as described in “Access and Error Log Messages” on page 138.

Starting RealServer Automatically

Once you have RealServer running satisfactorily, you can configure it to start automatically each time you start your computer.

RealServer is installed as a service under Windows NT. This means that it can be controlled from the Services Control Panel and starts and stops automatically when the system is booted or shut down.

When you run RealServer as a Service, errors are written to the Windows NT error logs rather than the error logs specified in the RealServer configuration file. You can view them just like any other Windows NT errors.

To uninstall RealServer as a service, run the **delsvc** program from the **bin** directory. Make sure that RealServer is stopped prior to removing it.

Starting RealServer on a UNIX Computer

Starting RealServer Manually

Because RealServer runs on a high-numbered, unprivileged port, you do not need super-user privileges to start it unless you have configured for PNAviaHTTP by setting HTTPPort 80. However, if you do start it while you are logged on as super-user, then RealServer can configure itself to use additional system resources, such as file descriptors, that it needs to support a large number of users connected simultaneously.

After you start RealServer with super-user privileges and it adjusts its resource limits, RealServer assumes the user and group IDs entered into the configuration file.

To start RealServer manually:

1. Change to the directory where you installed RealServer.
2. Start RealServer by entering:

```
bin/pnserver server.cfg
```

RealServer returns the command prompt and runs in the background. It does not return any messages to indicate that it has started.

If RealServer does not start, review the error messages in the RealServer Log as described in “Access and Error Log Messages” on page 138.

Starting RealServer Automatically

Once you have RealServer running satisfactorily, you can configure it to start automatically each time you start your computer.

Add the command to start RealServer to the boot-time scripts of your UNIX system. The boot-time scripts generally reside in files or directories beneath the **/etc** subdirectory. Be sure to use complete path names in your script.

If you do not have permission to change the boot-time scripts on your computer, you may need to have your system administrator do this for you.

Stopping RealServer

Shutting Down Gracefully

To shut down RealServer gracefully, you can prevent new connections without disconnecting current users. After your current users have disconnected, stop the RealServer.

To prevent new users from connecting to UNIX Servers without using System Manager:

1. Change the PnaPort configuration setting to an unused value such as 9999.
2. Issue the SIGHUP signal.

To prevent new users from connecting to any Server using System Manager:

1. Using the System Manager, connect to the Server.
2. Change the PnaPort configuration setting to an unused value such as 9999.

Be sure to change the PnaPort back to its normal value before restarting RealServer.

You can use the System Manager to check how many users are logged on to RealServer. See “Monitoring Performance” on page 160.

Stopping RealServer

To stop RealServer, follow the platform-specific directions below.

Windows NT

1. If you are running RealServer from the command line, press Ctrl+C. If you are running RealServer from the Server Control Center, from Server, select **Stop**.
2. If you are running RealServer as a Service, start Services Control Panel.
3. Select **PNServer**.
4. Click the **Stop** button.

UNIX

1. Log on either as super-user or by using the same user ID as RealServer.
2. If you know the process ID, type:

```
kill <processid>
```

If you don't know the process ID, change to the **pnservice** directory and type:

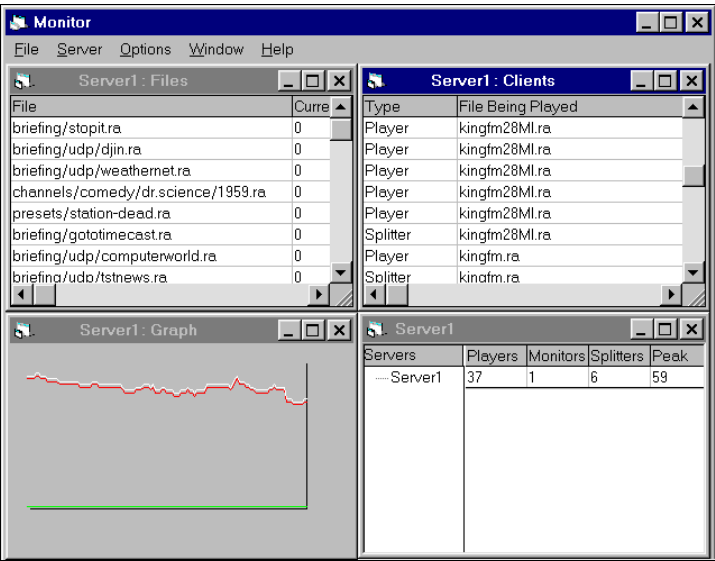
```
kill `cat logs/pnservice.pid`
```

Monitoring Performance

System Manager on Windows 95 or Windows NT lets you view the performance of one or more RealServers graphically. System Manager can monitor a Server running on any platform.

Monitoring Performance from Windows

System Manager includes windows that display clients currently connected, files being played, and a graph of all connections for the past two minutes. The System Manager for Windows is shown in the following figure:



This highly versatile tool lets you choose how you want connection information interpreted and displayed. For example, you can leave System Manager open on a corner of your screen with just the Player connections shown as a graph, giving you a visual sense of the connection activity on your RealServer.

1. **Windows 95 and NT 4.0:** Click the **Start** button, point to **Programs**, point to **RealAudio**, and click **RealAudio System Manager**.

Windows NT 3.51: Double-click the RealAudio program group and double-click the **RealAudio System Manager** icon.

2. On the File menu, click **Open**.
3. Click the name of the Server that you want to monitor and click the **OK** button.
4. On the Server menu, click **Clients**, **Files**, or **Graph** to display the type of information you want.

Clients Window

You can view the following information in the Clients window:

Column	Description
Type	The type of client connected: Player, Monitor (System Manager), or Encoder.
File Being Played	Name of the file being played from your RealServer.
Domain Name	The domain name or IP address of the client computer. To toggle between IP address and domain name, check Do DNS Lookups on the Clients tab of the Options dialog box.
Elapsed Time	The length of time that the client has been connected to that file since the System Manager has been attached to the Server. This information is also available in the access log.

Use the Clients window in System Manager to determine how many clients connect to your site simultaneously. You can multiply this number by 10 to 20 Kbps to determine how much bandwidth your RealServer is using.

If you want the Clients window to update continuously, select **Preferences** from the Options menu, select the **Client View** tab and check the **Update Continuously** box.

Files Window

The Files window tells what files are being accessed and the number of times each file is being played. This helps you determine which files are most and least popular, which could help you decide what new files to add or remove to improve the popularity of your site.

Column	Description
File	Name of the file currently being played.
Current	Number of clients currently connected to that file.
Total	Total number of connections made to this file since the System Manager was started.

If you want the Files window to update continuously, select **Preferences** from the Options menu, select the **File View** tab and check the **Update Continuously** box.

Graph Window

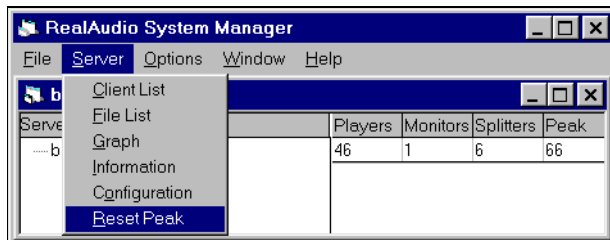
The Graph window gives a graphical interpretation of selected connections made to your Server in the past two minutes. To control what information appears on the graph, select **Preferences** from the Options menu, select the **Graph View** tab and check the boxes for the statistics you want to display.

Resetting the Peak Usage

The **Peak** value in the System Manager display is maintained until you restart the Server or manually reset the value.

To reset the Peak value:

- Click **ResetPeak** on the Server menu.



Network Performance Considerations

A number of factors can interfere with the quality of the media being delivered over the Internet. Media packets can be lost during delivery if they pass through slow routers or if the network is especially busy. Recurrent problems may indicate that you need to modify your connection to your Internet service provider.

To monitor media quality, read the connection statistics in the access log to learn more about packets that are early, late, missing, or out-of-order. Also, you should periodically use RealPlayer to listen to the clips on your RealServer. Open the Statistics window on RealPlayer and monitor the percentage of packet loss that is occurring. If the media quality you experience is poor, it is likely that your users are also experiencing poor media quality.

If you determine that there is a high level of packet loss, consult your Internet provider. You may need a faster Internet connection or there may be other problems with your Internet service.

Command Reference

This section contains reference information for commands used with RealServer. All but two of the commands (**kill** and **rssm**) work on both Windows NT computers and UNIX computers (**kill** and **rssm** work only on UNIX computers.)

The following commands are described:

Command	Description
cevents	Synchronized Multimedia compiler
kill (UNIX only)	Used with -HUP flags to force RealServer to reload with new configuration settings
pnserver	RealServer
raconv	Bandwidth negotiation file converter
rafile	Live File Creation Utility
rssm (UNIX only)	Command-line System Manager
slta	Simulated live transfer agent

For information on the **mpaste**, **rmcut**, **rmedit**, **rmttools**, **rmfile**, **rvslta**, **racut**, **rapaste**, and **rax** commands, which are used to edit media files, see “Editing Audio Files” on page 262 and “Editing Video Files” on page 202.

cevents

Name

cevents - Synchronized Multimedia compiler

Syntax

```
cevents <inputfile> <outputfile>
```

Description

The **cevents** utility takes the supplied text file containing the multimedia event descriptions and converts it to a compiled events file.

The following option is available:

-v Displays the version information of the **cevents** utility.

File Format

Inputfile has the following format:

```
u <starttime> <endtime> <URL>
```

where each entry is on a single line with each value separated by spaces and where:

`starttime` is the start time of the event in HH:MM:SS.t format

`endtime` is the end time of the event in HH:MM:SS.t format

`URL` is the URL of the page for the event

See Also

“Cevents” on page 165

kill

Name

kill

Syntax

```
kill -HUP <processID>
```

Description

You can reconfigure a running server on a UNIX machine using the command-line interface. First, change the parameters you want by editing the **server.cfg** file. Then use the kill command with the -HUP flags. This forces the Server to reload with the new configuration settings.

ProcessID is the process id of RealServer.

If you do not know the process id, run ps to obtain it. The parameters for ps depend upon the version of UNIX you are using:

UNIX platform	Command
BSDI, FreeBSD, LINUX, SunOS	ps -aux grep pns
AIX, HP-UX, DEC UNIX, IRIX, SOLARIS	ps -ef grep pns

pnserver

Name

pnserver - RealServer

Syntax

```
pnserver [-v] [-n] [-p port] <configfile>
```

Description

The **pnserver** command starts RealServer.

The following options are available:

- | | |
|---------|---|
| -v | Displays the version information of RealServer. This includes the platform, build and release tags used to identify a particular release. |
| -n | (UNIX only) Do not detach from the command terminal. This prevents the server from becoming a daemon process. |
| -p port | Use the supplied TCP port as the connection port for the server. This overrides any configuration file setting. |

configfile Specifies a file of configuration settings for **pnserver**. If no file is specified, uses the settings in **server.cfg**. If another file is specified, settings in this file override values in **server.cfg**. For information on configuration settings, see “Editing the Configuration File” in the 70.

raconv

Name

raconv - Bandwidth negotiation file converter

Syntax

```
raconv [-v] [-f] <file names> directory
```

Description

The **raconv** utility takes the supplied files and converts them to the Bandwidth Negotiation naming scheme and places them in the specified directory. More than one file name can be supplied.

The following option is available:

-v	Displays the version information of the utility. This includes the platform, build and release tags used to identify a particular release.
-f	Forces raconv to overwrite existing files with the same filenames.

Examples

The file **newband.ra** is encoded in three formats: RealAudio 2.0 - 28.8, RealAudio 3.0 - ISDN Mono, and RealAudio 3.0 - Dual ISDN Stereo. The files are stored in the following locations:


```
/usr/rawdata/old28_8/newband.ra  
/usr/rawdata/isdnmono/newband.ra  
/usr/rawdata/dualisdn/newband.ra
```

Issue the following commands:

```
raconv /usr/rawdata/old28_8/newband.ra /usr/rafiles  
raconv /usr/rawdata/isdnmono/newband.ra /usr/rafiles  
raconv /usr/rawdata/dualisdn/newband.ra /usr/rafiles
```

The **raconv** utility program creates a directory named **/usr/rafiles/newband.ra** that contains the files 28_8.36, dnet.50, and dnet.100.

rmfile

Name

rmfile - Live RealMedia File Creation Utility

Syntax

```
rmfile [-v] [-b] [-h-][-f config] [-t time] [-s size]  
-p password -e bandwidth URL destination
```

Description

The **rmfile** utility program creates files from live broadcasts. Use **rmfile** to archive live broadcasts for playback later.

The following options are available:

- | | |
|----|--|
| -v | Displays the version and copyright information for rmfile . |
| -b | Creates files using the bandwidth-negotiation style of file naming. For information on bandwidth negotiation, see “Bandwidth Negotiation” on page 223 of the Video Content Creation Guide. |

-h	Displays the syntax and options of the rmfile command
-f config	Forces rmfile to take its configuration information from the file named in config, instead of from the command line. This file uses the same format as the server.cfg file, but contains settings that pertain to rmfile only.
-t time	Specifies the amount of video to store in one file, by time. Use m for minutes, h for hours, and d for days. This option is mutually exclusive with the -s option.
-s size	Specifies the amount of video to store in one file, by size of file. Size is measured in megabytes (MB). This option is mutually exclusive with the -t option.
-p password	Specifies the password required to connect to RealServer broadcasting the video.
-e bandwidth	Specifies the bandwidth of the file being saved. Valid values are 14_4.18 , dnet.20 , dnet.25 , 28_8.36 , dnet.50 , and dnet.100 . The default value is set by the BandwidthEncoding configuration setting.
url	The URL for the live file being broadcast.
destination	The directory or filename in which to save the audio. If destination is a directory, rmfile uses the name of the live stream as the basis for creating filenames. If destination is a filename, rmfile uses the filename as the basis for creating filenames.

Example

To create a file every 30 minutes and use the filename **myvideo.rm**, enter:

```
rmfile -t 30m pnm://server:7071/live1.rm
myvideo.rm
```

Successive files are named myvideo0.rm, myvideo1.rm, and so on.

To save 10 MB files in the /usr/archive directory, enter:

```
rmfile -s 10 pnm://server:7071/live1.rm  
/usr/archive
```

Successive files are named live10.rm, live11.rm, live12.rm, and so on.

rafile

Name

rafile - Live RealAudio File Creation Utility

Syntax

```
rafile [-v] [-b] [-h-][[-f config] [-t time] [-s size]  
-p password -e bandwidth URL destination
```

Description

The **rafile** utility program creates files from live broadcasts. Use **rafile** to archive live broadcasts for playback later.

The following options are available:

- | | |
|----|--|
| -v | Displays the version and copyright information for rafile . |
| -b | Creates files using the bandwidth-negotiation style of file naming. For information on bandwidth negotiation, see “Bandwidth Negotiation” on page 271 of the Audio Content Creation Guide and on page 223 of the Video Content Creation Guide sections of this manual. |

-f config	Forces rafile to take its configuration information from the file named in config, instead of from the command line. This file uses the same format as the server.cfg file, but contains settings that pertain to rafile only.
-t time	Specifies the amount of audio to store in one file, by time. Use m for minutes, h for hours, and d for days. This option is mutually exclusive with the -s option.
-s size	Specifies the amount of audio to store in one file, by size of file. Size is measured in megabytes (MB). This option is mutually exclusive with the -t option.
-p password	Specifies the password required to connect to RealServer broadcasting the audio.
-e bandwidth	Specifies the bandwidth of the file being saved. Valid values are 14_4.18 , dnet.20 , dnet.25 , 28_8.36 , dnet.50 , and dnet.100 . The default value is set by the BandwidthEncoding configuration setting.
url	The URL for the live file being broadcast.
destination	The directory or filename in which to save the audio. If destination is a directory, rafile uses the name of the live stream as the basis for creating filenames. If destination is a filename, rafile uses the filename as the basis for creating filenames.

Example

To create a file every 30 minutes and use the filename **concert.ra**, enter:

```
rafile -t 30m pnm://server:7071/live1.ra  
concert.ra
```

Successive files are named concert0.ra, concert1.ra, and so on.

To save 10 MB files in the /usr/archive directory, enter:

```
rafile -s 10 pnm://server:7071/live1.ra  
/usr/archive
```

Successive files are named live10.ra, live11.ra, live12.ra, and so on.

Note **Rmfile** is the main utility and supports .rm files; .ra files were used in the RealAudio 3.0 Server.

rssm

Name

rssm - Command-line System Manager (also called System Monitor)

Syntax

```
rssm [-v] [-l <update>] [-p <password>] [-c] [-i] [-k]  
<hostname[:port]>
```

Description

System Manager enables remote monitoring and administration of RealServer from the UNIX command line. To connect System Manager to a Server, set hostname to the DNS name or IP address of the Server. If the Server is running on a port other than 7070, specify the port number.

The following options are available:

- | | |
|-------------|---|
| -v | Displays the version information of the System Manager. This includes the platform, build and release tags used to identify a particular release. |
| -l update | Sets the update period for output to the screen to update seconds. |
| -p password | Provides the password required by System Manager to connect to the Server. If this option is not used the System Manager prompts for the password. This |

feature is not secure. The password is easily accessible to knowledgeable searchers. The password is required each time you want to start monitoring a Server. You can include `-p <password>` in the **rssm** command line in automatic monitoring scripts to avoid having to enter the password interactively.

- c Connects to the server to verify it is still accepting connections and then exits. Prints a message if the connection fails and the exit status is non zero.
- i Starts interactive mode and permits entry of the commands listed in the command section.
- k Does DNS lookups on incoming IP addresses to translate them to full domain names. This command can slow down responses on System Manager. If you are experiencing delays in System Manager information or in response to commands, make sure that this feature is turned off.

Interactive Commands

After starting System Manager in interactive mode by using the `-i` command-line option, you can enter any of the following interactive commands at the System Manager prompt (`>`):

Command	Function
c	Displays the current configuration after it has been retrieved using the t command.
e	Resets peak usage value.
g	Displays the time that the peak usage value was last reset.
h or ?	Prints a list of commands.
i	Prints the Server's version number and platform.

Command	Function
k	Begins collating hostname information for connected clients by doing reverse DNS lookups on the IP numbers provided by the Server.
l	Provides the current list of connected clients.
n	Modifies a Server configuration variable.
o	Prints # of Players, System Managers, unknowns, and total connections to STDOUT every five minutes, or the number of seconds specified by the -l option on the command line. This command can be toggled to start and stop.
p	Prints Server license information.
s	Prints a single line of summarized status information.
u	Continuous display. Updates whenever a client status changes.
x	Exit the program.

Notes

System Manager can monitor a Server running on any platform. Information provided by System Manager includes the number and status of Player connections, System Manager connections, Unknown connections (connections currently being negotiated with the Server), and Total connections. This information can then be used to monitor activity on RealServer on a regular basis.

System Manager runs in two modes: interactive and non-interactive. When the System Manager is in the non-interactive mode, information is automatically appended to STDOUT every 5 minutes, unless that time span is modified by the -l command. The System Manager accepts commands from the command line; however, it does not prompt you.

The interactive mode is started with the `-i` command, which enables the System Manager to print prompts and accept commands from the command line.

System Manager displays the information about clients connected to the Server in the following format:

```
<client>  <name>
```

Where:

`<client>` is the type of client connected (Monitor or Player).

`<name>` is the domain name or IP address of that client.

For example, a client listing might look like:

```
monitor 204.71.154.93
Player  204.71.153.24
```

If you prefer to receive System Manager information in a report, use the `-l` option and append the output to a file. To do this, use the following command:

```
rssm -l<seconds> <hostname>[:port] >>
monitor.txt
```

Where:

`<seconds>` is the number of seconds between reports.

`<hostname>` is the name of the computer you are collecting data from.

`monitor.txt` is the name of the report that the information is appended to.

Example

To monitor a Server in interactive mode, with updates every 20 seconds and fully qualified host names for clients, use the following command:

```
rssm -l 20 -k -i yourServer:7070
```


slta

Name

slta - RealAudio simulated live transfer agent

Syntax

```
slta [-v] [-h] [-l] [-f config] [-p password]  
inputfile URL
```

Description

The **slta** utility delivers a stored RealAudio file as if it is a live event. This can be used to provide a test or delayed broadcast of a live event.

The following options are available:

- | | |
|-------------|---|
| -v | Prints out the version information of slta . This includes the platform, build and release tags used to identify a particular release. |
| -h | Displays the command syntax and options for the slta command. |
| -l | Continuously loops the input file. |
| -f config | Forces slta to take its configuration information from the file named in config, instead of from the command line. This file uses the same format as the server.cfg file, but contains settings that pertain to slta only. |
| -p password | Specifies the password slta uses to connect to RealServer; this value is specified by the Server's EncoderPassword configuration parameter. |
| inputfile | The RealServer file to be converted to a live RealServer file. |

URL Specifies the address of the output, exactly as it is used by RealPlayer to access the stream in the format **pnm://host/file.ra[:port]**. If you do not include a port number, **slta** uses port 7070.

Example

To play the file **ford01.ra** as a live file, enter:

```
slta -p fakeit ford01.ra pnm://server.com/car.ra
```

Encoding RealVideo Clips

RealVideo Encoder enables you to compress video files or input from a video device into one or more RealVideo formats. Output can be sent to a file or directly to RealServer for live broadcasting. You can select input files by browsing from within the Encoder, or you can drag-and-drop files for automated processing. The Encoder window displays information about input and output file formats and has fields for entering descriptive information.

This chapter details RealVideo Encoder operations, including supported input data formats, output options, and how to achieve the best possible audio and video quality.

Some video editing programs can encode and write files in RealVideo format. For example a RealVideo Encoder Plug-in is available for Adobe Premiere.

There is a difference between encoding from a live source and broadcasting a live event. Live encoding requires special license purchased with your RealServer. For information about serving a live stream refer to “Delivering Live Content” on page 281.

System Requirements

For information on system requirement for RealVideo Encoder, refer to “RealVideo Encoder Requirements” on page 25.

Source Files

RealVideo content may be created either from previously recorded digital video files or from an external video source. The Encoder does not support

compressed input files. Use a third-party editing utility to convert non-supported formats to a supported format.

The following inputs are supported:

- AVI
- QuickTime

AVI Files

RealVideo Encoder accepts compressed or uncompressed format (it is recommended that you use uncompressed formats).

The following are AVI requirements:

- The AVI file must have a color depth of 24-bits.
- The standard video frame size is 176 x 144.
- Height and width must both be divisible by 16 or can be 160 x 120.
- Indeo drivers must be installed on your machine for the encoder to be able to open the AVI. Typically, if you captured the AVI on the same machine as the encoder, the encoder has no problem opening it. Otherwise the Indeo drivers are available from Intel.

To determine if you have the AVI video codec for a particular AVI file:

1. Right-click the file name.
2. Click **Properties**.

3. Click the **Details** tab. Look at the Video Format field.



If the Video Format field says "Unknown format" most likely the corresponding AVI video codec is not installed.

4. If the Video Format field lists a format, click the **Preview** tab.



You should be able to view the video in the preview window.

Note If you can preview the video, and still receive an error when encoding the AVI file, the file may not be in 24-bit color. You can convert the file to 24-bit file using a third-party video editing tool.

QuickTime or .MOV files

RealVideo Encoder for Windows accepts uncompressed 24-bit RGB QuickTime video files containing 8- or 16-bit mono and stereo audio. RealVideo Encoder for Macintosh also accepts compress QuickTime files.

Image Size

RealVideo Encoder supports any size image that is a multiple of sixteen pixels. The most standard size is 176 pixels by 144 pixels. An image size of 160 x 120 is also supported.

Producing High Quality Video

To create the best possible streaming video, you must start with the best possible source material. Different video formats yield different qualities when digitized. Because RealVideo compression algorithms are lossy, some of the information contained in your original input is not included in the reconstructed signal sent to RealPlayer.

The common video formats in order of quality are:

1. Betacam-sp, also known simply as Beta. This format is common among video production professionals.
2. Laserdisc
3. S-VHS or Super-VHS
4. VHS

Satellite television services (e.g. Direct TV) have extremely high quality video. Their feed quality typically exceeds that of Laserdisc.

Video playback devices commonly have two types of video outputs, S-video and composite. S-video produces better results.

Live Capture-to-File

If you have a real-time capture station, you can use it to capture and compress directly into RealVideo format. To do this, use your live feed or the output of a video player (Beta, S-VHS, Laser Disc, etc.) and set the encoder to capture to file instead of capture to a live server feed. This has the advantage of eliminating the need to create and store intermediate AVI files which are very large and take considerable disk space. It is also the fastest way to capture for things like breaking news clips when time-to-post is important.

Editing

Limited video editing tools are available with RealVideo. For more information, refer to “Editing Video Files” on page 202. For more advanced editing features the following programs are recommended:

- Adobe Premiere
- In:sync Kohesion

RealVideo Encoder installs an Adobe Premiere plug-in, allowing you to encode directly from Premiere.

Producing High Quality Audio

- Use high quality source files.
- When possible, digitize the sound to a supported file format. Then pre-process the file with a sound editing program. Set the amplitude of your input signal to maximize the use of the available dynamic range.
- Eliminate any DC offset either while recording content or later with an audio editor. This removes low frequency noise.
- Use a CD quality sampling rate (44.1 kHz), sampling width (16-bit) and two channels when creating an input file that you intend to encode using multiple audio codecs. To ensure that the audio stays synchronized with video or other time critical media, use the following sampling rates for your source audio:

Codec	Sampling Rate
14.4	8, 16 or 32 kHz
15.2 Kbps voice	8, 16 or 32 kHz
6.5 Kbps voice	8, 16 or 32 kHz
8.5 Kbps voice	8, 16 or 32 kHz
8 Kbps music	8, 16 or 32 kHz
12 Kbps music	8, 16 or 32 kHz
16 Kbps music High response	11.025, 22.05 or 44.1 kHz
16 Kbps music Med response	11.025, 22.05 or 44.1 kHz
16 Kbps music Low response	8, 16 or 32 kHz
20 Kbps music stereo	8, 16 or 32 kHz
40 Kbps music mono	11.025, 22.05 or 44.1 kHz
40 Kbps music stereo	8, 16 or 32 kHz
80 Kbps music mono	11.025, 22.05 or 44.1 kHz
80 Kbps music stereo	8, 16 or 32 kHz

- The source files should contain signals of the maximum allowable amplitude. If the full amplitude range is not used, the resulting RealVideo files may sound flat. Adjust the range using a sound editor before encoding the file. Some sound editors have a **Normalize** function that maximizes levels automatically.
- If your original audio file signal exceeds the acceptable amplitude range, the file may contain “clipping.” Clipping can give rise to clicks or pops on playback. If your source file contains a clipped signal, your final RealVideo file will have high-frequency background noise or static.
- When encoding live-source audio, you have less opportunity to manipulate your input signal. Be sure that volume levels are prepared and tested. If you are not doing a live broadcast, you may want to record your input as a .wav or .au file so that you may digitally edit it prior to compression.
- Cut any unnecessarily long silences from the beginning or end of the output file to conserve space.

Encoding Templates

Before you begin encoding, you must make decisions about the appropriate settings for different types of audio and video input. Progressive Networks has supplied several pre-defined encoding templates to assist users in making those decisions. You can select from the pre-defined templates, adjust those templates or define a new template for the type of content you are encoding. For a list of templates, refer to “Pre-Defined Templates” on page 185. For information about creating templates, refer to “Creating Templates” on page 198.

Each template is optimized for a particular type of audio and video content as well as for bandwidth. Select one or more templates that best suit your needs.

RealServer can deliver clips encoded using multiple templates. In this way, you can reach the widest possible audience while still providing high-bandwidth users with the best listening experience. Using **Bandwidth Negotiation**, you can configure your site to automatically serve the appropriately encoded file.

For more information about Bandwidth Negotiation, refer to “Bandwidth Negotiation” on page 223 for RealVideo, and on page 271 for RealAudio.

Pre-Defined Templates

Below are the default settings for some common content types found in RealVideo Encoder. If one of the following templates does not achieve the effect you are looking for, try creating a new template by basing it on a pre-defined template and then modifying the settings.

Note It is recommended that you don't modify the pre-defined templates. If you want to change some of the settings, save the template as a new template. This way, you don't overwrite the pre-defined templates.

Template Name	Audio Codec	Video Bit Rate (Kbps)	Total Bit Rate (Kbps)	Video Quality	Frame Rate (fps)
High Action 28.8 w/ Music	8 Kbps music	11.0	19	100	.25
High Action 28.8 w/ Voice	6.5 Kbps voice	12.5	19	100	.25
Music Video 28.8, Emphasize Audio	12 Kbps music	7	19	100	.25
Music Video 28.8, Emphasize Video	8 Kbps music	11	19	100	Optimize
Talking Heads 28.8	6.5 Kbps voice	12.5	19	100	Optimize
High Action 56 (Fractal)	8.5 Kbps voice	36.5	45	70	10
High Action 56 w/ Music	16 Kbps music, High response	29	45	100	Optimize
High Action 56 w/ Voice	8.5 Kbps voice	36.5	45	100	Optimize
Music Video 56, Emphasize Audio	16 Kbps music, High response	29	45	100	Optimize
Music Video 56, Emphasize Video	12 Kbps music	33	45	100	Optimize
Talking Heads 56	8.5 Kbps voice	36.5	45	100	Optimize
High Action, 112, (Fractal)	12 Kbps music	78	90	70	15
LAN 250	40 Kbps music mono	210	250	100	Optimize
LAN 500	80 Kbps music mono	420	500	100	Optimize

*Target Bandwidth indicates the minimum bandwidth necessary to play the file:

28.8 corresponds to a 28.8 Kbps modem

56.0 corresponds to an ISDN or 56 Kbps modem

112 corresponds to a Dual ISDN or T1 line

To prevent pitch shifting in audio resampling the following sample rates need to be used with the following codecs:

Codec	Sample Rates
14.4	8, 16 or 32 kHz
28.8	8, 16 or 32 kHz
6.5	8, 16 or 32 kHz
8.5	8, 16 or 32 kHz
Dolby 8 Kbps	8, 16 or 32 kHz
Dolby 12 Kbps	8, 16 or 32 kHz
Dolby 16 Kbps High response	11025, 22050 or 44100 Hz
Dolby 16 Kbps Med response	11025, 22050 or 44100 Hz
Dolby 16 Kbps Low response	8, 16 or 32 kHz
Dolby 20 Kbps stereo	8, 16 or 32 kHz
Dolby 40 Kbps	11025, 22050 or 44100 Hz
Dolby 40 Kbps stereo	8, 16 or 32 kHz
Dolby 80 Kbps	11025, 22050 or 44100 Hz
Dolby 80 Kbps stereo	8, 16 or 32 kHz

RealVideo Encoder Plug-in for Adobe Premiere

RealVideo Encoder Plug-in for Adobe Premiere 4.2 exists for both Windows and Macintosh platforms. Currently, the Windows version is installed by RealVideo Encoder. The Windows Adobe Premiere plug-in is installed in the /Premiere/plugins directory. The Macintosh Adobe Premiere plug-in must be copied to the plug-in directory in the Premiere folder.

The RealVideo Encoder Adobe Premiere plug-in does not work with Adobe Premiere LE because Premiere LE does not support third party plug-ins.

Supported File Formats

RealVideo Encoder Plug-in for Adobe Premiere supports all of the file types supported by Adobe Premiere.

Movie File Formats	Audio File Formats
Video for Windows (.avi)	Audio Interchange (.aif)
QuickTime for Windows (uncompressed) and Macintosh (.mov) (compressed and uncompressed)	Windows Waveform (.wav)
FilmStrip (.flm)	

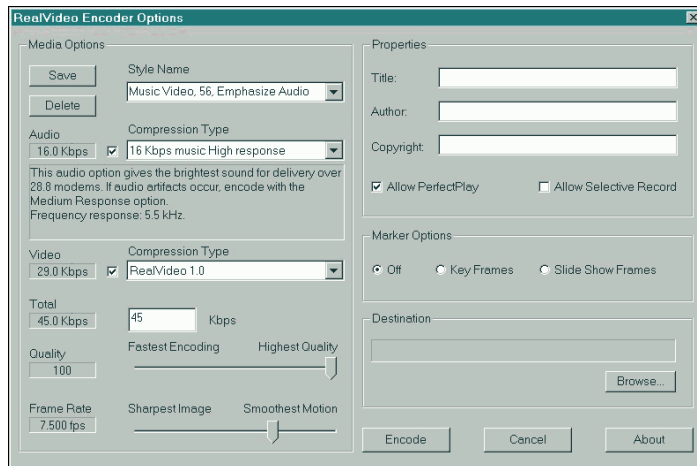
Audio source files must be sampled at 11, 22, or 44 kHz.

8-bit and 16-bit mono and stereo audio files are supported.

Encoding with Adobe Premiere 4.2

To encode audio and video files in Adobe Premiere 4.2:

1. Start Adobe Premiere 4.2.
2. Click **Open** from the File menu.
3. Select one of the supported file types. The file appears in the Clip window.
4. Select the Clip window.
5. Click **Export** from the File menu and select RealVideo Encoder. RealVideo Encoder Options window appears (Windows version):



6. In the media options pane, select one of the pre-installed styles. These styles are guidelines for creating content containing both audio and video. You can save your own styles, or delete styles that you do not use. If you delete a style you created yourself, it will be permanently deleted. If you delete a style which was installed by RealVideo Encoder, you can get it back by reinstalling RealVideo Encoder.
7. Once you have selected a style, you can change any of the settings for the audio codec, the video codec, the total bit rate, the Quality, or the Frame Rate.

8. In the properties pane, enter the Title, Author, and Copyright information for your output file. These fields are optional.
9. RealPlayer users with low bandwidth modems can experience files encoded for a higher bandwidth by partially downloading audio data before beginning playback. If you want to allow this, click the **Allow PerfectPlay** checkbox. This is the default setting.
10. If you want to allow RealPlayer Plus users to save your clip to disk, click the **Allow Selective Record** checkbox.
11. In the Marker Options frame, click the appropriate option. Refer to Marker Options for more information.
12. In the destination pane, click **Select** and enter the name of the output file.
13. Click the **Encode** button. Adobe Premiere begins encoding the file.
14. When the file is encoded, a Status dialog box appears, providing details about the encoded file. Click the **OK** button.
15. Play the encoded file using RealPlayer.

Marker Options

Marker options allow you to take advantage of the Marker feature in Adobe Premiere.

- Click **Off** to encode the entire clip regardless of markers. Choosing this option means that you don't have control over which frames get encoded.
- Click **Key Frames** to encode a marked file. Choosing this option will ensure that all frames that are marked are encoded.
- Click **Slide Show Frames** to encode only the marked frames. Choosing this option allows you to create slide shows out of video files and gives you control over exactly which frames you want to use.

Refer to your Adobe Premiere documentation for information on how to set and clear numbered and unnumbered markers. The plug-in recognizes both Premiere's numbered (up to 10 available) and unnumbered markers (up to 1000 available).

Using RealVideo Encoder for Windows and Macintosh

RealVideo Encoder for Windows can do static and live encoding. Whether or not you will be able to do live encoding depends on the license key for your RealServer. RealVideo Encoder for Macintosh does static encoding only.

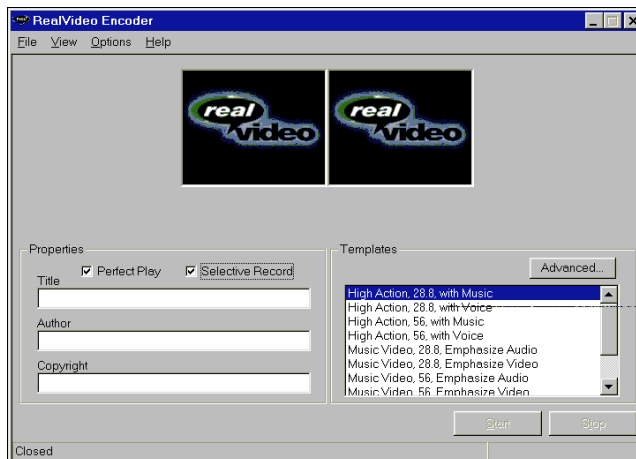
RealVideo Encoder converts video in common file formats to RealVideo format. You specify the input source and output file, and you can specify options such as the compression type and copyright string to be included in the output file. For a list of supported input formats, see “Source Files” on page 179.

Progressive Networks has supplied several pre-defined encoding templates to assist users in determining the appropriate settings for different types of video input. You can select a pre-defined template or define a template specifically for the type of video you are encoding. For more information, refer to “Encoding Templates” on page 185 and “Creating Templates” on page 198.

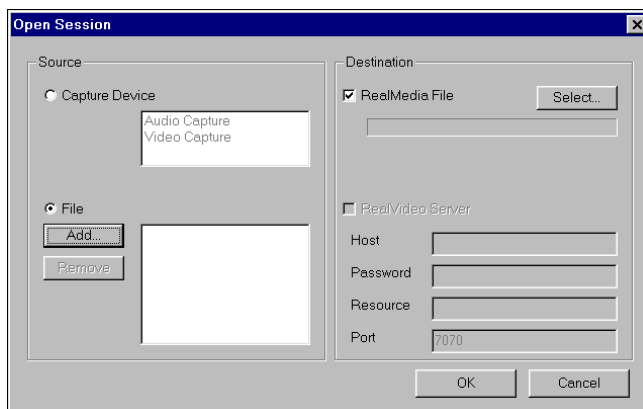
Note If you have the appropriate license key for RealServer, RealVideo Encoder can deliver live content for broadcasting live events. For information about RealVideo Live Encoder for Windows, refer to “RealVideo Live Encoder for Windows,” on page 288.

To encode a file using RealVideo Encoder for Windows:

1. Start RealVideo Encoder. RealVideo Encoder window opens:



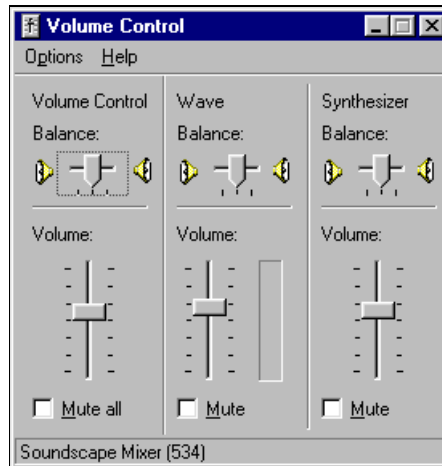
2. Click **Open Session** from the File menu. The Open Session window appears:



3. In the source pane, verify that **File** is selected.
4. Click the **Add** button. The Add Source File dialog box appears.
5. Select the file you want to encode. Click the **Open** button.
6. In the destination pane, verify that a check appears in **RealMedia File** checkbox.

7. Click the **Select** button. The Select Destination File dialog appears.
 8. Enter the destination file name.
 9. Click the **Save** button.
 10. Click the **OK** button. The Open Session window closes.
 11. In the properties pane, enter the Title, Author, and Copyright information for your output file. These fields are optional.
 12. If you want to allow RealPlayer users with low bandwidth modems to experience RealVideo files encoded for a higher bandwidth by partially downloading video data before beginning playback, click the **PerfectPlay** checkbox.
 13. If you want to allow RealPlayer and RealPlayer Plus users to save your RealVideo signal to disk, click the **Selective Record** checkbox.
 14. In the templates pane, select the templates with which to encode the file. For more information, refer to “Encoding Templates” on page 185.
- If you want to create your own template, click the **Advanced** button. Refer to “Creating New Templates” on page 200.

15. If you want to control the volume, select **Volume Control** from the Options menu. The Volume Control window displays. You can adjust the volume using the controls.



16. If you want to crop the image, select **Crop Input Image** from the Options menu. The Image Cropping window displays. Notice the crop lines around the image. These lines show the portion of the image that will be encoded. Use the Left, Top, Width, and Height up and down arrows to adjust the size and location of the crop lines.



17. Click the **Start** button to start the encoding process. When the file is encoded, an Encoding Complete message appears indicating the bit rate achieved.

Note If RealVideo Encoder is unable to encode the file within the selected bit rate, a message window appears. If the final bit rate is not acceptable to your needs, you need to re-encode the file with a different template.

18. View the encoded file with RealPlayer.

Drag-and-Drop Encoding for Windows

RealVideo Encoder for Windows supports drag-and-drop encoding:

- Click an input-video-file icon and drag it onto an open Encoder window. This enters path and file name information into the appropriate Encoder fields. Then, you only need to enter the descriptive information and settings and click the **Start** button to begin the encoding process.

Multi-Template Encoding and Bandwidth Negotiation

You can encode more than one RealMedia file at the same time using multi-template encoding. Encoding a single file using multiple templates generates multiple output files, either for testing or to allow the server to stream the best file for the user's stated bandwidth.

To encode a file with multiple templates:

1. Click on the **template name** to select the first template.
2. Control-click on a second **template name**. Repeat this step until you have selected all the templates you want to use to encode.
3. Click the **Start** button. The Multiple Template Encoding dialog box displays with a list of the templates it will use to encode.
4. Verify that the **Enable Bandwidth Negotiation** box is checked. The files will automatically be named for the server.

5. Click the **Begin** button. The encoding process starts.
6. To view the status of the encoding, double-click on the **template name**. The Template Summary dialog box displays.
7. When the encoded is finished, you will be prompted:

Multi-template encoding is completed.
8. Click the **OK** button.

Command Line Encoding for Windows

In some situations you may find it convenient to encode live or on-demand content within the DOS command line. The following syntax is used:

```
RVBatch RVEncode.exe <options>
```

where *options* are any of RealVideo Encoder options described below.

Option	Description (defaults in parenthesis)
/I	Use this option to specify an Input File
/O	Use this option to specify an outfile or dir - Output File Name or Directory (infile.rm or dir\YYYYMMDDHHMMSS.rm)
/L	Use this option to specify Use Live Input
/S	Use this option to specify "server[:port]/file" - Server Name, Port and File. (Port 7070)
/W	Use this option to specify password - Server Password
/D	Use this option to specify hhh:mm:ss - Maximum Encoding Duration (continuous)
/A	Use this option to specify an Audio Codec (0)
/V	Use this option to specify a Video Codec (0)
/F	Use this option to specify a framerate - Frame Rate. (Optimal)
/B	Use this option to specify Total Kbps for clip. (100)
/Q	Use this option to specify Quality 1-100. (100)
/T	Use this option to specify a Clip Title

Option	Description (defaults in parenthesis)
/U	Use this option to specify a Clip Author
/C	Use this option to specify a Clip Copyright
/P	Use this option to Enable Perfect Play. Valid options are: 0 Disabled 1 Enabled (1)
/R	Use this option to Enable Selective Record. Valid options are: 0 Disabled 1 Enabled (0)
/X	Use this option to Enable Audio Encoding. Valid options are: 0 Disabled 1 Enabled (1)
/Y	Use this option to Enable Video Encoding. Valid options are: 0 Disabled 1 Enabled (1)
/?	Use this option to Display the HELP information

Batch Encoding on Macintosh

Batch encoding on the Macintosh uses AppleScript. For more information, refer to:

www.real.com

Creating Templates

Each template is comprised of an audio codec, video bit rate, total bit rate, quality and frame rate. The quality of RealVideo encoding is affected by these settings and by the actual content. Below are content elements that can affect the quality of a video encoding at 19 Kbps:

Content Type	Expected Encoding Results
Fixed camera shot, low motion	Clean frames, with high frame rates
Fixed camera shot, medium motion	Clean frames, somewhat lower frame rates
Multiple camera shots, low motion	Clean frames with 1 fps or less
Zoom with low motion	Clean frames with moderate frame rate
Fixed camera shot, high motion	Clean frames with moderate frame rate
Multiple camera shots, high motion	Clean frames with less than 1 fps

Subjective results based on content elements

RealVideo encoder produces significantly better frames with higher frame rates when used with bandwidths higher than 28.8 Kbps.

There are three parameters that can be adjusted to produce RealVideo: audio codec, video bit rate, and frame rate. When deciding which parameters to set, the total bit rate must be considered. The total bit rate is the actual bit rate at which the AVI is encoded. If the target bit rate is 28.8 Kbps, then the total bit rate should be 19 Kbps. The following are the recommended total bit rates for common bandwidths.

Target Bit Rate	Total Bit Rate
28.8 Kbps	19 Kbps
56.0 Kbps	45 Kbps
64.0 Kbps	56 Kbps
128.0 Kbps	105 Kbps

Target and total bit rates

After selecting the total bit rate according to the table above, choose an audio codec.

Parameter	Bandwidth	Audio Codec
0	6500 bps	6.5 Kbps voice
1	8500 bps	8.5 Kbps voice
2	16000 bps	16 Kbps music Low response
3	16000 bps	16 Kbps music Medium response
4	16000 bps	16 Kbps music High response
5	20000 bps	20 Kbps music stereo
6	40000 bps	40 Kbps music mono
7	4000 bps	40 Kbps music stereo
8	80000 bps	80 Kbps music mono
9	80000 bps	80 Kbps music stereo
10	8000 bps	8 Kbps music
11	12000 bps	12 Kbps music
12	15200 bps	15.2 Kbps voice

Since RealAudio codecs have discreet bandwidths, the video bit rate is the difference between the total bit rate and the bit rate of the chosen audio codec:

$$\text{Video Bit Rate} = (\text{Total Bit Rate}) - (\text{Bit Rate of Audio Codec})$$

In RealVideo Encoder, the video bit rate is calculated automatically. There are two video codecs:

Parameter	Video Codec
0	RealVideo (Standard)
1	RealVideo (Fractal)

Next choose a frame rate. Not all content types support high frame rates for a given bandwidth. The highest frame rate that can be achieved at 19 Kbps is 7.5 frames per second (fps) for Talking Head type content. Different content types require different frame rates. For example, while Talking Heads at 19 Kbps might support 7.5 fps, music videos only support 1 fps or less.

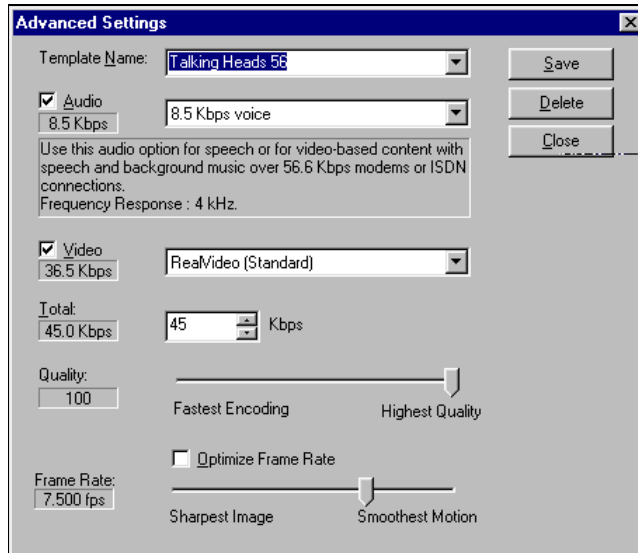
Note If you are getting high levels of latency, you are exceeding the bounds of your content type. To solve high latency:

- Choose optimize frame-rate. This lets the encoder adjust the frame rate.
- Reduce the screen size (176 x 144 is recommended). Use multiples of 16 in setting frame size.
- Lower the frame rate.

Creating New Templates

To create a new template:

1. In the templates pane, click the **Advanced** button. The Advanced Setting window appears:



2. In Windows, enter the name of the new template in the Template Name field. In Macintosh, select Custom and click the **Save** button. A dialog box will display. Enter the new template name and click the **OK** button.

Note It is recommended that you don't modify the pre-defined templates. If you want to change some of the settings, save the template as a new template. This way you don't overwrite the pre-defined templates

3. Select an audio codec.
4. Select a Video codec. RealVideo Encoder has two video codecs available, RealVideo (Standard) and RealVideo (Fractal). The bit rate of RealVideo 1.0 codec varies depending on the video codec and total bit rate selected.
5. Select the total bit rate for the encoded video file.
6. Select the Quality by moving the slider.

7. Select the Frame Rate by moving the slider. When using the RealVideo Standard video codec, you can select **Optimize Frame Rate** to let the encoder automatically select the frame rate.
8. Click the **Save** button to save the template.
9. Click the **Close** button to close the Advanced Settings window.

Editing Video Files

If you want to edit your video files, you have two options—you can edit your .rm files, or you can edit your original source files.

Editing Input Files

Editing the original source files is the editing method of choice. Encoded video is stored in indivisible clips of varying duration depending on the encoding algorithm and the video bit rate. This limits how precisely encoded video can be cut and pasted.

RealVideo Encoder is distributed with a Windows editing utility, RMTTools, and DOS: RMEdit, RMPaste and RMDump which enable simple editing of RealVideo files.

Editing RealVideo Files in Windows

RMTTools, the Windows 95 and Windows NT editing utility, is installed with RealVideo Encoder. This utility enables you to perform three basic editing tasks:

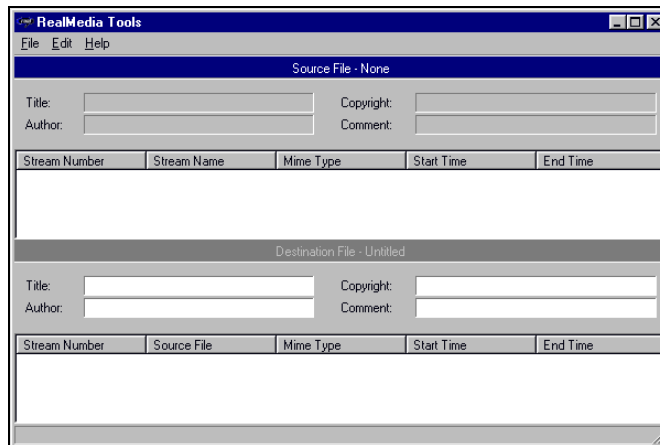
- Edit the file properties
- Examine the contents of a file
- Edit the data of a file.

Editing File Properties

RMTools allows you to edit the static information about the file: title, author, copyright, and comment; and the stream name and mime type for each individual stream. You can also enable or disable PerfectPlay or Selective Record. Editing the file properties does not affect the data.

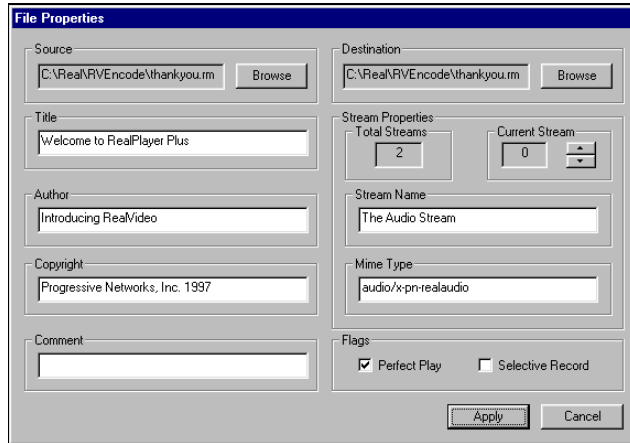
To edit file properties of RealVideo files using RMTools:

1. Click the **Start** button. From **Programs**, select **Real**.
2. Select **RealMedia Tools**. The RealMedia Tools window appears. Notice the window is divided into two panes, the source file pane on top and the destination file pane on the bottom.



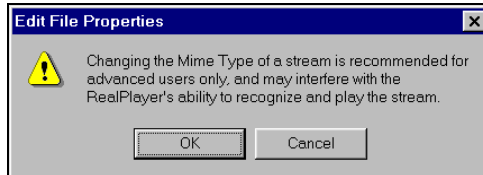
2. Select **Open Source** from the File menu. The Open File dialog box displays. Select the file you want to edit. The file properties information is displayed in the source file window. To sort by the stream attribute, click the column. You can also resize the columns by placing the cursor over the column border and then clicking-and-dragging.

3. Select **File Properties** from the Edit menu. The File Properties dialog box appears. The file name is displayed in the source box and in the destination box. New file names can be selected by clicking Browse and selecting a new file from the dialog box.



4. To save changes to the existing file, do not change the destination file name. To save the changes to a new file, click **Browse** to select or enter a different name in the destination box.
5. Type the new title name, author, copyright information, and any comment in the appropriate boxes. (You can also type a new title, author, copyright, or comment directly in the fields on the source and destination panes.)
6. The total number of streams is listed in the Stream Properties box. If you want to change the name or the mime type of one stream, select the stream to be changed by clicking the up or down arrow.
7. If you want to change the stream name, type the new name in the Stream Name box.

8. Changing the mime type is recommended for advanced users only. If you are certain you wish to change the mime type, type the new mime type in the Mime Type box. If you change the mime type, the following warning will appear after you have applied the changes.



9. RealPlayer users with low bandwidth modems can experience files encoded for a higher bandwidth by partially downloading audio data before beginning playback. If you want to allow this, check the **PerfectPlay** checkbox in the Flag box.
10. If you want to allow RealPlayer Plus users to save your clip to disk, check the **Selective Record** checkbox.
11. Click the **Apply** button. The changes are applied to the file and displayed in the source file pane of the RealMedia Tools window.

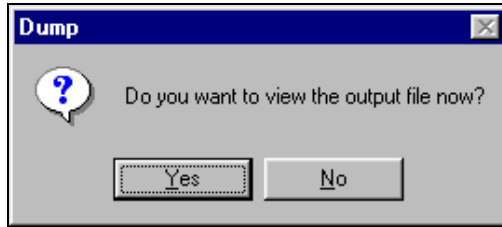
Examining the Contents of a RealVideo File

RMTools enables you to view the contents of an .rm file by dumping the contents to a text file. You can then edit or print the file as you would any other text file.

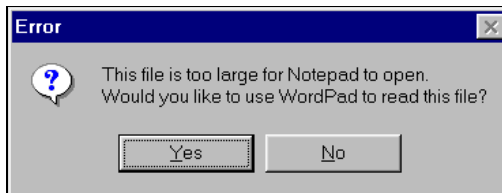
To examine the contents of a RealVideo file:

1. Select **New Session** from the **File** menu.
2. Open the Source file from the File menu. The file information is displayed in the source pane.
3. Select **Dump** from the File menu. The Dump As dialog box appears. The default name is the same as the source file name, with the extension changed to .txt. You may also select or enter another dump file name.

4. Click the **Save** button. You will be prompted to view the file now. Click the **Yes** button.



5. The following error message may appear. Click the **Yes** button. The Notepad or WordPad window displays the contents of the file. For an example of an .rm text file, refer to “Command Line Editing of RealVideo Files” on page 210.



6. You can view, edit, and print the file just as you would any other text file.

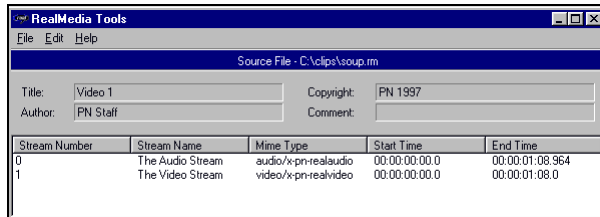
Editing RealVideo Data

RMTTools allows simple editing of .rm files. You can put multiple clips in one file; you can shorten the length of a clip; and you can take a video stream from one file and add the audio stream from another file. When working with more than one file, streams of the same mime type must have the same encoding parameters if they are going to be pasted together.

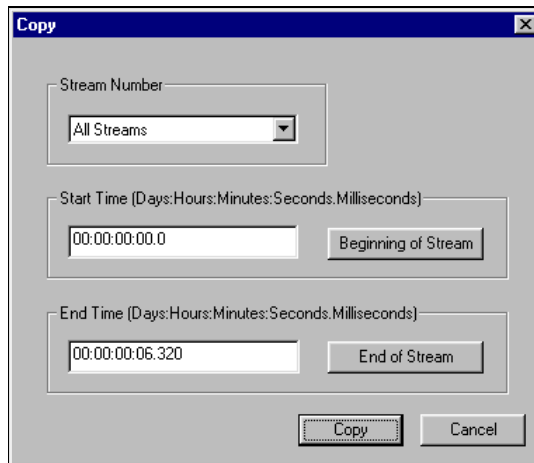
To place two RealVideo clips in one file:

1. Select **New Session** from the File menu.
2. Select **Open Source** from the File menu. The Open File dialog box appears.

3. Select the file you want to edit. Click the **Open** button. The source file pane displays the stream number, the stream name, mime type, and the start and end times.

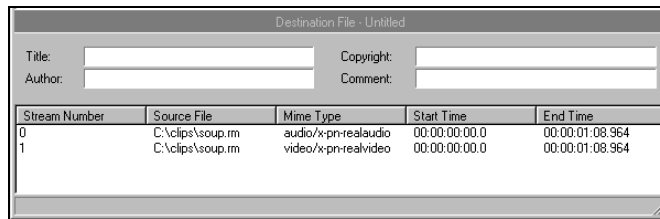


4. Select **Copy** from the Edit menu. The Copy dialog box appears.



5. In the Stream Number box, accept the default, **All Streams**.
6. Do not change the start and end times. Click the **Copy** button.
7. Select **Paste** from the Edit menu. The Paste dialog box appears.

8. Click the **Paste** button. The destination pane displays the stream number, source file, mime type, start time and end time.

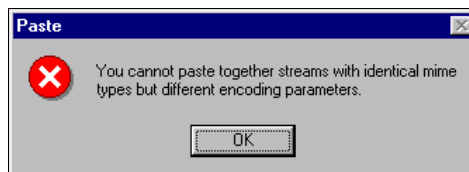


Destination File - Untitled

Title: Copyright:
 Author: Comment:

Stream Number	Source File	Mime Type	Start Time	End Time
0	C:\clips\soup.rm	audio/x-pn-realaudio	00:00:00:00.0	00:00:01:08.964
1	C:\clips\soup.rm	video/x-pn-realvideo	00:00:00:00.0	00:00:01:08.964

9. Right-click the source pane. A shortcut menu appears. Click **Open Source**. Choose a second clip to be added at the end of the first clip. (Both clips must have the same encoding parameters.)
10. Click the **Open** button. The source pane displays the information from the new file while the destination pane displays the information from the streams that were previously pasted.
11. Select **Copy** from the Edit menu.
12. Accept the default, **All Streams**, and the start and end times. Click the **Apply** button.
13. Right-click the destination pane. A shortcut menu appears. Click the **Paste** button. The Paste dialog box appears. The previous end time displays as the new start time.
14. Click the **Paste** button. The second clip is listed below the first clip.
15. If the following warning displays, you must select a different clip or re-encode the clip so that the encoding parameters are the same.



16. Select **Save As** from the File menu. The Save As dialog box appears.
17. Type the new file name. Click the **Save** button.
18. Use RealPlayer to view the new file.

To put the video stream of one file with the audio stream of another:

1. Select **New Session** from the **File** menu.
2. Select **Open Source** from the **File** menu. The Open File dialog box appears.
3. Select the first file you want to edit. Click the **Open** button. The source file pane displays the information from the file.
4. Select **Copy** from the Edit menu. The Copy dialog box appears.
5. In the Stream Number box, select the stream that contains the video. Click **Copy**.
6. Select **Paste** from the Edit menu. The Paste dialog box appears.
7. Click **Paste**. The destination pane displays the information from the new file. Notice only the video stream has been pasted into the destination pane.
8. Right-click the source file pane. A shortcut menu appears. Click **Open Source**.
9. Choose a second file with the same encoding parameters as the first clip, and of approximately the same duration. Click the **Open** button. The source pane displays the information from the file.
10. Right-click the source pane. A shortcut menu appears. Click **Copy**. The Copy dialog box appears.
11. In the Stream Number box, select the stream that contains the audio.
12. Do not change the start and end times. Click **Copy**.
13. Right-click the destination pane. A shortcut menu appears. Click **Paste**. The Paste dialog box appears.
14. Click **Paste**. The audio stream is listed below the video stream. Notice both streams have the same start time, and should have similar end times.

Stream Number	Source File	Mime Type	Start Time	End Time
1	C:\clip\shortsoup.rm	video/x-prerehvideo	00:00:00.0	00:00:36.206
0	C:\clip\vwdclip.rm	audio/x-prereaudio	00:00:00.0	00:00:32.0

15. Select **Save As** from the File menu. The Save As dialog box appears.
16. Type the new file name. Click the **Save** button.
17. Use RealPlayer to view the new file.

Command Line Editing of RealVideo Files

The RealVideo file editing tools, RMCut, RMPaste, RMEdit, and RMDump are run from a DOS command line. The RMCut tool is used to cut a specific portion of a RealVideo file without changing the original source. The RMPaste tool is used to combine two or more RealVideo files. The RMEdit tool is used to modify the title, author, copyright, comment, mime type, or stream name. It can also be used to print the current values for the file or stream. The RMDump tool is used to examine the contents of a RealVideo file.

Editing RealVideo Files

RMEdit has five separate forms. The first form modifies any or all of the title, author, copyright, and comment for the file.

```
rmedit -i <input> -o <output> [-t <title>]
[-a <author>] [-c <copyright>] [-C <comment>]
```

The second form modifies any or all of mime type and stream name for the selected stream.

```
rmedit -i <input> -o <output> (-S <stream number>
[-m <mimetype>] [-s <stream name>])
```

The third form prints out the current values for the file and for each stream that RMEdit is capable of modifying.

```
rmedit -i <input>
```

The fourth form sets or clears Perfect Play mode and Selective Record.

```
rmedit -i <input> -o <output> [-r [ON|OFF]] [-p  
[ON|OFF]]
```

The fifth form prints the RMDump version and copyright notice.

```
rmedit -v
```

where:

- <input> is the path and file name to the input file.
- <output> is the path to the output file that contains the edited file.
- <title> is the title text.
- <author> is the author text.
- <copyright> is the copyright text.
- <comment> is the comment text.
- <stream number> is the stream number for the following two parameters. (May be repeated for any number of streams.)
- <mimetype> is the mimetype for a specified stream.
- <stream name> is the stream name for a specified stream.
- v prints the RMDump version and copyright notice.
- p is the perfect play mode.
- r is the set/clears selective record.

Note Only valid mime types (those accepted by the player) should be entered in the mime type field.

Note Input and output file names must not be the same.

The following examples demonstrate the use of the RMEdit tool:

1. To create a new file modifying the title, enter:

```
rmedit -i original.rm -onewfile.rm -t "My New  
Title"
```

2. To change the stream name of stream 1, enter:

```
rmedit -i original.rm -onewfile.rm -S1 \  
-s "New Stream Name"
```

3. To display the current contents of the file, enter:

```
rmedit -i original.rm
```

Note Long file names are acceptable, but file names containing spaces must be contained in quotes.

Cutting RealVideo Files

Use the RMCut tool to cut a specific portion of a RealVideo file without modifying the original source. A cut is made by setting the start and end times for individual streams. Not entering a start time defaults to the beginning of the file and likewise, not entering an end time defaults to the end of the file. RMCut can be used without RMPaste to remove the front or end of a file.

To create a copy of a piece of a RealVideo file:

1. View your original RealVideo file to determine the start and end times of the segment you want to copy (you can read the times on the status bar of RealPlayer).

2. Create a copy by typing:

```
rmcut -i <input> ([-S <stream>] [-s <start>] \ [-e <end>]) -o <output> -v
```

where:

<input> is the path to the input file

<stream> is the stream number. If not specified, all streams are used. It can be repeated to grab one or more streams.

<start> is the start time in

Days:Hours:Minutes:Seconds.Milliseconds format. If missing, this parameter defaults to the beginning of the file.

<end> is the end time in Days:Hours:Minutes:Seconds.Milliseconds format. If missing, this parameter defaults to the end of the start+input length. (Multiple start end sequences can be used to cut discontinuous portions of the stream.)

<output> is the path to the output file that contains the edited file.

-v is the print version and copyright notice

Note An image map stream's start and end times must go from the beginning to the end of the file. Use RMCut to strip out the image map before working

with the RealVideo file. Then, to alter and re-merge the image map file with the video image, refer to “Image Maps” on page 218.

The following examples demonstrate the use of the RMCut tool.

1. To create a new file with only stream zero, enter:

```
rmcut -i source.rm -o cutfile.rm -S0
```

2. To create a new file with the first 30 seconds of stream 0:

```
rmcut -i source.rm -o cutfile.rm -s0.0 -e30.0
```

3. To create a new file with the first 30 second of stream 0 and with the first 45 seconds of stream 1, enter:

```
rmcut -i source.rm -o cutfile.rm \  
-S0 -s0.0 -S1 -s0.0 -e45.0
```

Examining the Contents of RealVideo Files

RMDump is used to examine the contents of a RealVideo file. RMDump accepts two input parameters: the input file and the output file.

To use RMDump, type:

```
rmddump -i <infile> -o <outfile>  
rmddump -v
```

Where:

<infile> is the input .rm extension file

<outfile> is the output text file (use a text editor to view)

<-v> prints the RMDump version and copyright notice.

An example of how to run RMDump is shown below:

Example

```
>rmddump -i input.rm -o out.txt  
dumping input.rm to out.txt . . .  
dumping realmedia headers . . .  
dumping stream packets  
dumping stream 0 . . .  
dumping stream 1 . . .  
done  
! Elapsed time = 556 milliseconds
```

The next step is to use an editor like Notepad on Windows NT/95 to examine the contents of the RMDump.

```
RealMedia File: input.rm
=====
RM_HEADER_OBJECT
... deleted to save space ...
=====
RM_PROPERTIES_OBJECT
... deleted to save space ...
=====
RM_CONTENT_OBJECT
... deleted to save space ...
=====
RM_MEDIA_PROPERTIES_OBJECT
object_id: 0x4D445052 (RPDM)
size: 155 bytes
object version: 0
stream_number: 0
max_bit_rate: 12062 bps
avg_bit_rate: 12062 bps
max_packet_size: 288 bytes
avg_packet_size: 288 bytes
start time: 0 milliseconds
preroll: 191 milliseconds
duration: 224898 milliseconds
stream_name: The Audio Stream
mime_type: audio/x-pn-realaudio
type_specific_length: 73
=====
RM_MEDIA_PROPERTIES_OBJECT
object_id: 0x4D445052 (RPDM)
size: 116 bytes
object version: 0
stream_number: 1
max_bit_rate: 7000 bps
avg_bit_rate: 7000 bps
max_packet_size: 668 bytes
avg_packet_size: 564 bytes
start time: 0 milliseconds
preroll: 3474 milliseconds
duration: 228000 milliseconds
stream_name: The Video Stream
mime_type: video/x-pn-realvideo
type_specific_length: 34
=====
RM_DATA_OBJECT
```

```
... deleted to save space ...
=====
Seek Time: 1 with reqtime = 0 and actualtime = 0
packet 0: stream 0 timestamp 0 size: 288 keyframe
packet 1: stream 1 timestamp 0 size: 639 keyframe
packet 2: stream 1 timestamp 0 size: 619 keyframe
packet 3: stream 1 timestamp 0 size: 599 keyframe
packet 4: stream 1 timestamp 0 size: 544 keyframe
packet 5: stream 1 timestamp 0 size: 639 keyframe
packet 6: stream 0 timestamp 192 size: 288 keyframe
packet 7: stream 0 timestamp 384 size: 288 keyframe
packet 8: stream 0 timestamp 575 size: 288 keyframe
packet 9: stream 0 timestamp 767 size: 288 keyframe
packet 10: stream 0 timestamp 958 size: 288 keyframe
packet 11: stream 0 timestamp 1150 size: 288 keyframe
packet 12: stream 0 timestamp 1341 size: 288 keyframe
packet 13: stream 0 timestamp 1533 size: 288 keyframe
packet 14: stream 0 timestamp 1724 size: 288 keyframe
packet 15: stream 0 timestamp 1916 size: 288 keyframe
packet 16: stream 0 timestamp 2108 size: 288 keyframe
packet 17: stream 0 timestamp 2299 size: 288 keyframe
packet 18: stream 0 timestamp 2491 size: 288 keyframe
```

The RM_MEDIA_PROPERTIES_OBJECT sections describe the stream type and stream number for each stream.

- The keyframe field is used to identify the start of a series of related packets.
- Keyframes are sometimes too large to form a single packet. In this case multiple keyframes will be seen with the same timestamps.
- Some codecs produce packets with the keyframe flag always set. And others produce a mixture of keyframe and non keyframe packets.
- Packet sizes can be identical in size or vary in size within a single stream.
- The RM_MEDIA_PROPERTIES_OBJECT sections contain attributes of each stream which include:
 1. The stream number.
 2. The stream type.
 3. The total stream type (audio or video).
 4. The total duration of the stream in milliseconds.
 5. The preroll which describes how long to load buffers into the player before playing the stream.

Combining RealVideo Files

RMPaste assumes that streams on the input side are joined to streams on the output side based upon their mime types. Start and End times are specified to map the input file, which may contain one or more streams, to a new time range in the output file. Only one file can use default start and end times. Start and end times must not overlap. If streams overlap, packets from both streams are intermingled based upon timestamp.

```
rmpaste (-i <input> [-s <start>] [-e <end>]  
[-f <from stream> -t <to stream>]) -o <output> -v
```

where:

<input> is the path to the input file. There can be one or more -i -s -e parameters sequences.
<start> is the start time in Days:Hours:Minutes:Seconds.Milliseconds format. If missing this parameter defaults to the beginning of the file.
<end> is the end time in Days:Hours:Minutes:Seconds.Milliseconds format. If missing this parameter defaults to start+input length.
<output> is the path to the output file that contains the edited file.
<from stream> is the input stream number to be sent to the to stream.
<to stream> is the output stream number.
-v is the print version and copyright notice.

Note RMPaste can paste in additional streams and add to existing streams. When streams are pasted together, the timestamps must not overlap.

Note If multiple streams are in the input file and the timestamps of the streams do not start at the same time, the start time applies to the earliest stream in the file. The start time plus the difference between the streams first packet is used for the remaining streams. See first example.

The following examples demonstrate the use of the RMPaste tool:

1. If you cut three streams from a source file using:

```
rmcut -i source.rm -o cutfile1.rm \  
-S0 -s10.0 -S1 -s20.0 -S2 -s30.0
```


resulting in an output file like:

Times

Stream0:	0.3	1.1
Stream1:	10.4	12.4
Stream2:	22.0	32.0

and mapped it using:

```
rmppaste -i cutfile1.rm -s5.0 \
-i cutfile2.rm -e5.0 \
-o paste.rm
```

the new stream times for cutfile1.rm would be:

Times

Stream0:	5.3	6.1
Stream1:	15.4	17.4
Stream2:	27.0	37.0

with cutfile2.rm placed from the beginning of the file to 5 seconds into the file. (packets not shown)

2. To create a new file using the first 60 seconds of the source.rm file swapping the first 45 seconds with the next 15 seconds in the file enter:

```
rmcut -i source.rm -o cutfile1.rm -s0.0 -e45.0
rmcut -i source.rm -o cutfile2.rm -s45.0 -e60.0
rmppaste -onewfile.rm -icutfile2.rm -s0.0 -e15.0 \
-icutfile1.rm -s15.0 -e60.0
```

3. To add a new stream to an existing file:

```
rmppaste -ioriginal.rm -inewstream.rm -onewfile.rm
```

Image Maps

"Click-able" image maps allow users to interact with the video content by adding image-specific video seeking and URL commands. Image maps are fully customizable; actions can be connected to rectangular, circular or polygonal content regions and can be varied over specific user defined time intervals.

To create Image Maps:

1. Create an Image Map File (text file).
2. Merge the Image Map file (with a .rm file).

Creating an Image Map File

Image Map Files are text files that have HTML-like tags which specify the location and action of each image map. Image map files should be created in a text editor and saved as text only.

All tags shown below are required unless otherwise stated. Negative values for numbers are not allowed.

DURATION Tag

This tag must be at the beginning of the file. The DURATION tag specifies the amount of time from the start time of the first map to the end time of the last map in the file. The format is as follows:

DURATION=x:x:x:x

Tag Name	Description
DURATION	Times are specified in the following format: Days:Hours:Minutes:Seconds:MilliSeconds. All of the these fields must be present. That means if you don't want to specify a particular setting, fill 0's in that space.

MAP Tag

This line describes the overall properties of the image map. The format is as follows:

<MAP START=x:x:x:x END=x:x:x:x COORDS=x,y,x1,y1>

Tag Name	Description
START	Times are specified in the following format: Days:Hours:Minutes:Seconds:MilliSeconds. All of the these fields must be present. That means if you don't want to specify a particular setting, fill 0's in that space. If START contains all 0's then the start of the clip is assumed.
END	Times are specified in the following format: Days:Hours:Minutes:Seconds:MilliSeconds. All of the these fields must be present. That means if you don't want to specify a particular setting, fill 0's in that space. If END contains all 0's then the end of clip is assumed.
COORDS	Specifies the rectangle that this entire map is going to occupy. If you enter an area bigger than the entire display area, the entire area is active. Any area not located within this rectangle will be clipped.

You may have as many MAP tags in a file as you wish.

AREA Tag

There can be 0 to 40 area lines in an image map. These lines describe the different regions that are active within an image map.

<AREA START=x:x:x:x END=x:x:x:x SHAPE=XXXX
COORDS=x0,y0,x1,y1,x2,y2,...xn,yn action_tag ALT="" >

Tag Name	Description
START	Specifies the start time of this area during the map. This field is optional; if you do not include a START time, the area begins at the beginning of the map time. Times are specified in the following format: Days:Hours:Minutes:Seconds:MilliSeconds. All of the these fields must be present. That means if you don't want to specify a particular setting, fill 0's in that space. If START contains all 0's then the start of the current map is assumed.
END	<p>Specifies the end time of this area during the map. This field is optional; if you do not include an END time, the area ends at the end of the map time. Times are specified in the following format: Days:Hours:Minutes:Seconds:MilliSeconds. All of the these fields must be present. That means if you don't want to specify a particular setting, fill 0's in that space. If END contains all 0's then the end of map is assumed.</p> <p>Note If the start and end times of an area are outside the start and end times of the map, then the start and end times of the map are used.</p>
SHAPE	<p>The XXXX for the SHAPE tag can be one of the following values: (CIRCLE, RECTANGLE, POLYGON).</p> <p>CIRCLE - COORDS tag should specify three values: centerX,centerY,radius</p> <p>RECTANGLE - COORDs tag should specify four values: left,top,right,bottom</p> <p>POLYGON - COORDs tag should specify at least 6 values (x and y coordinates for at least 3 points). These are the vertices for the polygon.</p>
action_tag	<p>The action_tag is one of the following: PLAYER, URL, SEEK.</p> <p>PLAYER - Used to specify a new stream to play with RealPlayer. For example: PLAYER="pnm://video.real.com/welcome.rm"</p> <p>URL - Used to specify a new URL to display in a browser. For example: URL="http://www.real.com"</p> <p>SEEK - Used to specify a time within the current clip to seek to. For example: SEEK=0:0:0:5:0.</p> <p>Note the format is just like the START and END tags.</p>
ALT	The ALT tag contains text that appears in the status bar of the player when the mouse is over this AREA and it is active. If you want no text then use "" as shown above.

/MAP Tag

This line means you are done with the current image map.

Example

```
DURATION=0:0:0:40:0
<MAP START=0:0:0:0:0 END=0:0:0:5:20 COORDS=0,0,100,100>
<AREA SHAPE=CIRCLE COORDS=50,50,10
    URL="http://www.real.com" ALT="Home Page">
</MAP>
<MAP START=0:0:0:5:20 END=0:0:0:20:3 COORDS=0,0,100,100>
<AREA SHAPE=RECTANGLE COORDS=0,0,50,50 SEEK=0:0:0:3:98
    ALT="Seek to a point in the clip">
</MAP>
<MAP START=0:0:0:20:3 END=0:0:0:40:0 COORDS=0,0,100,100>
<AREA END=0:0:0:30:0 SHAPE=POLYGON COORDS=0,50,50,0,100,50
    PLAYER="pnm://video.real.com/welcome.rm" ALT="">
<AREA START=0:0:0:31:0 SHAPE=POLYGON
    COORDS=0,25,25,0,100,50 SEEK=0:0:0:0:0 ALT="Rewind to
    Beginning">
</MAP>
```

Merging an Image Map file with a Video Image

When you have finished creating the Image Map file, you convert the Image Map text file into a .rm file using the RMMerge tool that is installed with RealVideo Encoder. Then, you merge that .rm file with an encoded video (.rm) file.

To convert the Image Map file into a .rm file:

1. From a DOS command line, change directories to the directory that RealVideo Encoder was installed, which by default is C:\REAL\RVENCODE
2. Enter the following command:

```
rmmerge -f rmimap.dll image_map_text image_map_rm
```

Done..... appears when the image map text file has been converted to a .rm file.

Note If you receive the message Error Parsing file, verify that the Image Map file is saved as text only and the tags are used correctly.

3. Merge the Image Map .rm file with a video (.rm) file by typing the following command:

```
rmmerge image_map_rm video_rm final_rm
```

The final_rm file contains your Image Map as well as the audio/video clip.

4. Verify the Image Map by playing the final_rm file with RealPlayer.

Modifying RealVideo File Descriptions

You can change the Title, Author, or Copyright text, and modify the Selective Record and PerfectPlay settings in a .rm file using several different methods:

Method	Description
RMTools	The is the editing tool available for Windows 95 and Windows NT. Refer to “Editing File Properties” on page 202.
RMEdit Tool	This is a command line tool available for Windows. Refer to “Command Line Editing of RealVideo Files” on page 210.
.ram File	Set the Title, Author, and Copyright text strings in the .ram file. Refer to “Metafiles” on page 327.

Bandwidth Negotiation

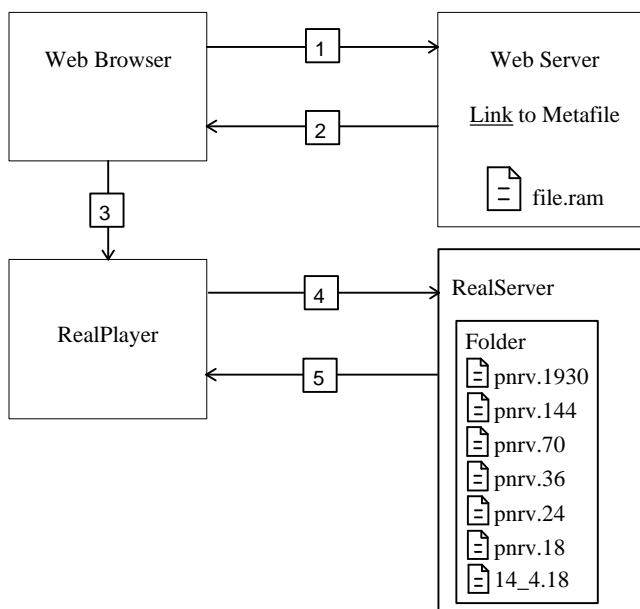
You can configure RealServer to deliver files encoded with different algorithms based on the bandwidth capability of the user's RealPlayer. The Player gets the best quality the connection can handle without having to explicitly choose among multiple links and Web pages. You can choose to provide as many versions of each file as you want. The bandwidth negotiation process is transparent to users.

Bandwidth negotiation requires only one link on your Web site to a particular clip. To provide content in multiple formats without bandwidth negotiation, your Web site must have a separate hypertext link and metafile for each format.

For information on using Bandwidth Negotiation with Synchronized Multimedia presentations, refer to "Using Synchronized Multimedia with Bandwidth Negotiation" on page 354.

The following diagram shows how RealPlayer and RealServer determine which file to play based on bandwidth.

File organization is the key to bandwidth negotiation. This figure shows a single link on a Web page, and seven available RealVideo files encoded using different algorithms.



The following steps correspond to the numbers in the figure:

1. The user clicks a link to a metafile on a Web page.
2. The Web server returns the metafile to the Web browser and based on the **.ram** file extension, sets the MIME type of the metafile to **audio/x-pn-RealAudio**.
3. The Web browser looks up the MIME type of the metafile, starts RealPlayer as a helper application, and passes it the metafile.
4. RealPlayer reads the first URL from the metafile and requests it from RealServer. Based on its preference settings, RealPlayer also sends a list of RealVideo compression types it supports.
5. RealServer checks the directory specified by the URL and begins streaming the highest bandwidth file supported by RealPlayer.

The following table shows the metafile for each bandwidth and connection for RealVideo clips.

Connection	Bandwidth (bps)	File Name
T1	1544000	pnrv.1930
Dual ISDN	115200	pnrv.144
Single ISDN	56700	pnrv.70
28.8	28800	pnrv.36
19.2	19200	pnrv.24
14.4	14400	pnrv.18

The following table shows the file formats that RealAudio Player version 1.0 through 3.0 requests and in what order they are requested:

Player Version	Bandwidth Setting	Files Requested
1.0	All	14_4.18
2.0	14.4	14_4.18
2.0	28.8, ISDN, T1	28_8.36 14_4.18
2.1	14.4	14_4.18
2.1	28.8, ISDN, T1	28_8.36 14_4.18
3.0	14.4	14_4.18
3.0	28.8	dnet.25 dnet.20 28_8.36 14_4.18
3.0	ISDN	dnet.50 dnet.25 dnet.20 28_8.36 14_4.18
3.0	T1	dnet.100 dnet.50 28_8.36 dnet.25 dnet.20 14_4.18

Note RealAudio Player 3.0 with a 28.8 Kbps connection always plays RealAudio 2.0 - 28.8 (28_8.36) format if it is available. If you want to provide another 28.8 format such as 28.8 Stereo (dnet.25), do not also provide the 28_8.36 format. If you do not provide any format supported by a Player, that Player receives a message to upgrade to the current Player release.

The following table shows the file formats that RealPlayer 4.0 requests from RealAudio servers 1.0 through 3.0, and RealServer in the order they are requested. The unshaded file formats can be played in real time. The file formats shaded in gray can be played, but the player will use a buffered playback mechanism to play the file.

T1/LAN	Dual ISDN	ISDN	28.8	19.2	14.4
pnrv.1930	pnrv.144	pnrv.70	pnrv.36	pnrv.24	pnrv.18
pnrv.144	dnet.100	dnet.50	dnet.25	dnet.20	dnet.15
dnet.100	pnrv.70	pnrv.36	pnrv.24	28_8.36	sipr.12
pnrv.70	dnet.50	dnet.25	dnet.20	pnrv.18	dnet10
dnet.50	pnrv.36	pnrv.24	28_8.36	dnet.15	sipr.10
pnrv.36	dnet.25	dnet.20	pnrv.18	sipr.12	1pcJ.18
dnet.25	pnrv.24	28_8.36	dnet.15	dnet10	28_8.36
pnrv.24	dnet.20	pnrv.18	sipr.12	sipr.10	dnet.20
dnet.20	28_8.36	dnet.15	dnet10	1pcJ.18	pnrv.24
28_8.36	pnrv.18	sipr.12	sipr.10	dnet.25	dnet.25
pnrv.18	dnet.15	dnet10	1pcJ.18	pnrv.36	pnrv.36
dnet.15	sipr.12	sipr.10	dnet.50	dnet.50	dnet.50
sipr.12	dnet10	1pcJ.18	pnrv.70	pnrv.70	pnrv.70
dnet10	sipr.10	dnet.100	dnet.100	dnet.100	dnet.100
sipr.10	1pcJ.18	pnrv.144	pnrv.144	pnrv.144	pnrv.144
1pcJ.18	pnrv.1930	pnrv.1930	pnrv.1930	pnrv.1930	pnrv.1930

Bandwidth Negotiation Example

In this example, you deliver one of three RealVideo formats depending on the connection speed and RealPlayer version.

To setup this example:

1. Encode the source file in the following formats:
 - RealMedia 19 Kbps Video - 6.5 Kbps Audio
 - 45 Kbps Video - 16 Kbps Audio
 - 100 Kbps Video - 40 Kbps Stereo

2. Create a metafile named **myvideo.rm** containing a URL such as:

```
pnm://www.real.com/music/myvideo.rm
```

3. Create a link to the metafile in a Web page. The following HTML code is a typical link:

```
<A HREF="http://www.real.com/myvideo.rm">  
Watch myvideo</A>
```

4. On RealServer computer, create a directory named **myvideo.rm** in the **/music** directory.
5. In this directory, store the three RealAudio files, renamed **pnrv.24**, etc. as shown in the previous table.

The file that is played depends on the Player connection and version:

- ISDN or faster connection with RealPlayer 3.0 or later: RealAudio 3.0 - ISDN Stereo format (**dnet.50**)
- 28.8 Kbps connection with RealAudio Player 3.0 or later: RealAudio 3.0 - 28.8 Stereo format (**dnet.25**)
- 14.4 Kbps connection with RealPlayer 3.0 or later: RealAudio 2.0 - 14.4 format (**14_4.18**)
- 14.4 Kbps or faster connection with RealAudio Player version 2.1 and earlier: RealAudio 2.0 - 14.4 format (**14_4.18**)

RealVideo Content Creation for Internet Appliances and Set-top Boxes

Recently, there has been a significant increase in the number of appliances such as televisions, Internet phones etc. that are able to view content resident on servers on the World Wide Web. Increasingly, these appliances will also have the capability to view and play RealAudio and RealVideo content. While these devices may be able to play RealAudio or RealVideo, they are often constrained in various ways. Some examples of these constraints include:

- Their Internet connections are at relatively low bit rates.
- They may not have the capability to playback RealVideo.
- They may be able to playback content that uses only the RealVideo(Standard) codec.

To make your content accessible by this increasingly large user population, we strongly recommend creating content in a variety of formats that are playable both on RealAudio/RealVideo players running on PCs and workstations as well as appliances. To do so, you can take advantage of RealVideo's many codec offerings and do one or more of the following:

- If you choose to use the RealVideo(Fractal) codec to create high-quality, high bandwidth content, you may also wish to use the RealVideo(Standard) codec to create a lower bit-rate version of the same content. The same strategy could be used if you are creating high bandwidth content using RealVideo(Standard).
- Create links on your web-pages to audio only versions of video presentations. These audio only versions can be created at multiples bandwidths (using RealVideo bandwidth negotiation). You may also wish to create content using codecs that are part of RealAudio 3.0 to accommodate users that may have audio players.
- Another alternative would be to use bandwidth negotiation and create high bandwidth RealVideo content and low bandwidth RealAudio versions of the same content.

These options are well supported in the RealVideo encoder. As time progresses, making these options available on your site will only add to your audience.

Encoding RealAudio Clips

RealAudio Encoder enables you to compress audio files or input from a sound device into one or more RealAudio formats. Output can be sent to a file or directly to RealServer for live broadcasting. You can select input files by browsing from within the Encoder, or drag and drop files onto the Encoder icon for automated processing. The Encoder window displays information about input and output file formats and has fields for entering descriptive information. (UNIX users work in a command line environment.)

This chapter details RealAudio Encoder operations, including supported input data formats, output options, and how to achieve the best possible sound quality.

Some sound editing programs can encode and write files in the RealAudio format. These include Macromedia Sound Edit 16 (Macintosh), Sonic Foundry Sound Forge, and Syntrillium CoolEdit.

There is a difference between encoding from a live source and broadcasting a live event. RealAudio Encoder included with RealServer can deliver live content for broadcasting live events. For information about serving a live stream refer to “Delivering Live Content” on page 281. The free RealAudio Encoder available from Progressive Network’s Web site does not support live broadcasting.

System Requirements

For information on system requirement for RealAudio Encoder, refer to “RealAudio Encoder Requirements” on page 28.

Source Files

RealAudio content may be created either from previously recorded digital audio files or from an external audio source. The Encoder does not support compressed input files. Use a sound-editing utility to convert non-supported formats to a supported format.

The following inputs are supported:

Type	Platform	Sample Size
.wav audio	Windows, UNIX	8- or 16-bit mono or stereo
.au audio	Windows, UNIX, Macintosh	8-bit μ or stereo or 16-bit linear mono or stereo
.pcm audio	Windows, UNIX	8- or 16-bit mono or stereo
.snd audio	Windows, Macintosh	8- or 16-bit mono or stereo
.sd2 audio	Macintosh	8- or 16-bit mono or stereo
.aiff audio	Macintosh	8- or 16-bit mono or stereo
Live feed	Windows Live, Mac Live, UNIX Live	8- or 16-bit mono or stereo

Note You can use a stereo input file to produce a mono or a stereo output file. However, you cannot use a mono input file to produce a stereo output file.

The valid sampling rates are: 8 kHz, 11.025 kHz, 16 kHz, 22.05 kHz, and 44.1 kHz.

Producing High Quality Sound

The quality of your RealAudio clips depends on the quality of the input source. Because the RealAudio compression algorithms are lossy, some of the information contained in your original audio input will not be included in the reconstructed signal sent to RealPlayer. You produce higher-grade audio following compression/decompression if you start with a high-fidelity recording with full dynamic range and a high signal-to-noise ratio.

The following is some advice for producing high-quality source files:

Sound Cards

- Recording with direct digital input achieves the best quality sound. This can be achieved if you have a professional CD-ROM or Digital Audio Tapes (DATs) player with digital output and a sound card with digital input. If you do not have this setup, use the highest quality sound card possible.

Creating Input Files

- Use a sound editor that lets you modify the settings and control the attributes of the file; for example, Macromedia Sound Edit 16 (Macintosh), Sonic Foundry Sound Forge, Sound Wave Designer, and Syntrillium CoolEdit.
- Use high quality source files from compact disks (CDs) or DATs.
- When possible, digitize the sound to a supported file format. Then pre-process the file with a sound editing program. Set the amplitude of your input signal to maximize the use of the available dynamic range.
- Eliminate any DC offset either while recording content or later with an audio editor. This removes low frequency noise.
- Use a CD quality sampling rate (44.1 kHz), sampling width (16-bit) and two channels when creating an input file. You can always downsample and convert to one channel later.
- The source files should contain signals of the maximum allowable amplitude. If the full amplitude range is not used, the resulting RealAudio files may sound hollow. Adjust the range using a sound editor before

encoding the file. Some sound editors have a **Normalize** function that will maximize levels automatically.

- If your original audio file signal exceeds the acceptable amplitude range, the file may contain “clipping.” Clipping can give rise to clicks or pops on playback. If your source file contains a clipped signal, your final RealAudio file will have high-frequency background noise or static.
- When encoding live-source audio, you have less opportunity to manipulate your input signal. Be sure that volume levels are prepared and tested. If you are not doing a live broadcast, you may want to record your input as a .wav or .au file so that you may digitally edit it prior to compression.
- Cut any unnecessarily long silences from the beginning or end of the output file to conserve space.

Choosing an Encoding Algorithm

When you encode an audio file, you select an encoding algorithm. RealAudio Encoder can encode using several different algorithms. Each encoding algorithm is optimized for a particular type of audio and connection speed bandwidth. You need to select one or more algorithms that best suit your needs.

It is possible to offer more than one encoding algorithm from your RealServer. In this way, you can reach the widest possible audience while still providing high-bandwidth users with the best listening experience. Using **Bandwidth Negotiation**, you can configure your site to automatically serve the appropriately encoded file.

For more information about Bandwidth Negotiation, refer to “Bandwidth Negotiation” on page 271.

The following algorithms are available:

Encoding Algorithm	Description	File Size per Second
RealAudio 2.0 - 14.4	Use for speech over 14.4 Kbps modems. Frequency response: 4.0 kHz	1 KB
RealAudio 2.0 - 28.8	Use for voice content and voice with background music over 28.8 Kbps modems. This option increases intelligibility under high packet loss conditions. Frequency response: 4.0 kHz	1.8 KB
RealAudio 3.0 - 28.8 Mono, full response	This option gives the brightest sound for delivery over 28.8 Kbps modems. If audio artifacts occur, encode with the medium response option. Frequency response: 5.5 kHz	2 KB
RealAudio 3.0 - 28.8 Mono, medium response	This option may improve clarity for music with snare drums, cymbals and vocals. If audio artifacts occur, encode with the narrow response option. Frequency response: 4.7 kHz	2 KB
RealAudio 3.0 - 28.8 Mono, narrow response	This option may improve clarity for speech-intensive music or noisy signals. Frequency response is somewhat reduced. Frequency response: 4.0 kHz	2 KB
RealAudio 3.0 - 28.8 Stereo	Use for general stereo content. Frequency response: 4.0 kHz	2.5 KB

Encoding Algorithm	Description	File Size per Second
RealAudio 3.0 - ISDN Mono	Use for general mono content over ISDN connections. Frequency response: 11.0 kHz	4.9 KB
RealAudio 3.0 - ISDN Stereo	Use for stereo content over ISDN connections. Frequency response: 8.0 kHz	4.9 KB
RealAudio 3.0 - Dual ISDN Mono	Use for optimal quality for mono content. Frequency response: 20.0 kHz (CD Quality)	9.8 KB
RealAudio 3.0 - Dual ISDN Stereo	Use for optimal quality for stereo. Frequency response: 16.0 kHz (Broadcast Quality)	9.8 KB

Encoding RealAudio files with Sound Forge

Sound Forge by Sonic Foundry provides excellent sound editing tools as well as the ability to save files in RealAudio format. This is a four step process:

- Recording the file
- Processing the file
- Setting the RealAudio text fields
- Saving the file in RealAudio format

Sonic Foundry has recently updated Sound Forge 4.0 and the Batch Converter Plug-In to fully support RealAudio technology. The update is available to all Sound Forge 4.0 users on Sonic Foundry's Web site: <http://www.sfoundry.com>

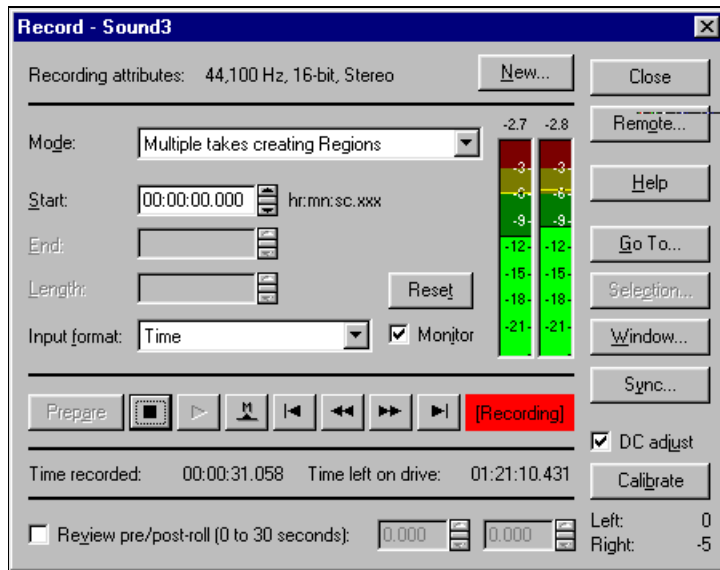
Sound Forge enables you to batch encode files easily. Refer to “Batch Encoding with Sound Forge” on page 241 for more information.

Note For more information about using the Sound Forge application, refer to the Sound Forge documentation.

To record a file in Sound Forge:

1. Click the **Record** button. The Record window appears displaying the settings from the last clip recorded.
2. Use the defaults or click the **New** button to change the Sample Rate, Sample Size or number of channels of the source file.
3. Check the **Monitor** box and play the clip to be encoded. Check for clipping. You want the levels to be mostly in the green area, a little in the yellow and only occasionally in the red. If Sound Forge displays a red **Clip** warning above the Monitor, you should:
 - a) Stop the input source.
 - b) Click the **Reset** button.
 - c) Lower the volume of the input source either by turning down the output of your sound source or by turning down the input volume on your sound card’s mixer window. In general, you don’t want any of your levels too high or too low.
 - d) Check the levels again. It is better to be conservative with the starting levels.
4. Check the **DC Adjust** box. Sound Forge can automatically calculate how much offset is necessary. With your sound source connected but no sound playing, click the **Calibrate** button. If you always use the same input device, you only have to perform this step once. Otherwise, you should recalibrate each time you switch input devices.

- Click the **Record** button on the Record window to start the record process.



- When the file is recorded completely or when you click the **Stop** button, a graphical display of the wave file appears.
- Click the **Close** button to close the Record window.

To prepare a recorded file for RealAudio encoding:

When you have finished recording a file, it is recommended that you perform the following processes on the file before saving the file in a RealAudio format.

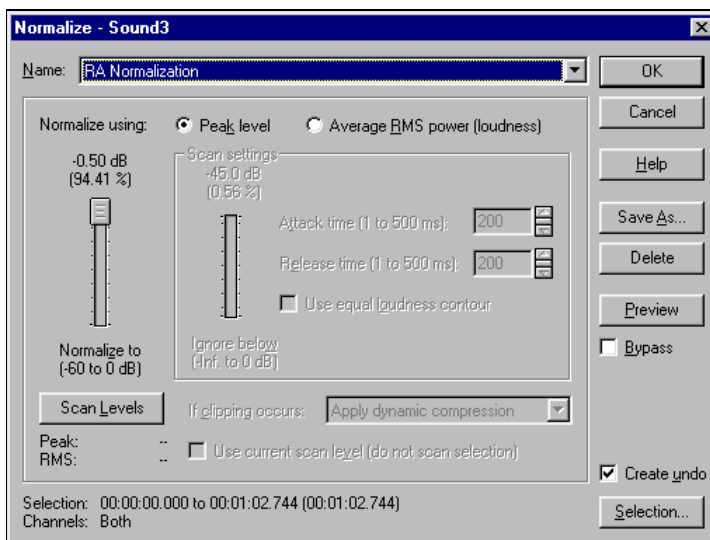
Note Sound Forge allows you to save settings for each of these processes. So once you set them up the first time, you can always use the same settings. Refer to the Sound Forge documentation for information on setting presets.

Normalize

This process ensures that the loudest part of the file is at the maximum recording level.

- Press Ctrl-A to select the entire file.

2. Click **Normalize** from the Process menu. The Normalize window appears:



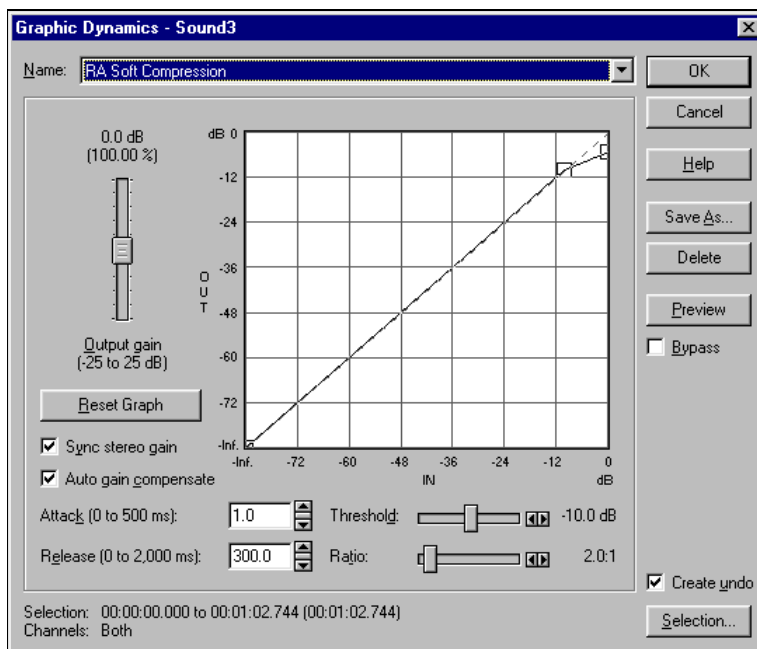
3. Click the **Peak Level** box.
4. Move the **Normalize to** slider to **-50 dB (94.41%)**.
5. Click the **OK** button. Sound Forge normalizes the file.

Compress

If you have not already audio compressed the clip, you can do so now. RealAudio files benefit from compression because it increases the overall volume and intelligibility of the sound. The suggested settings use very light compression.

1. Press Ctrl-A to select the entire file.

- Click **Dynamics** from the Effects menu and select **Graphic**. The Graphic Dynamics window appears:

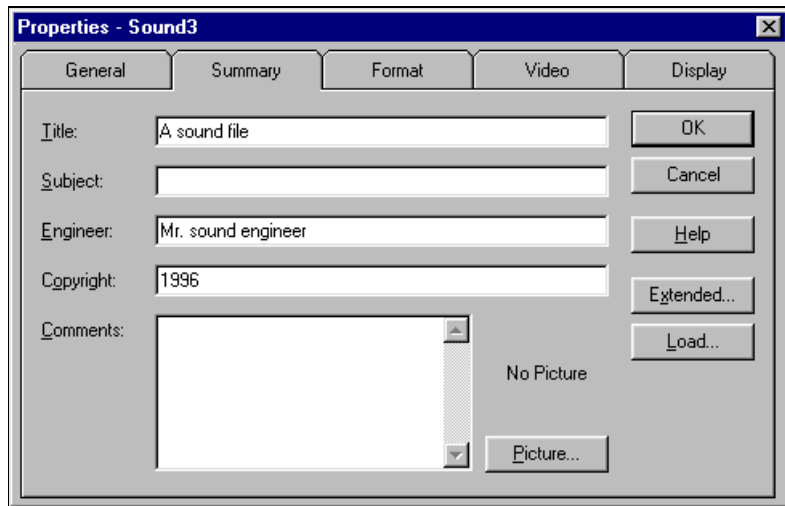


- Check the **Sync Stereo Gain** box.
- Check the **Auto Gain Compensate** box.
- Move the **Threshold** slider to **-10.00 dB**.
- Move the **Ratio** slider to **2.0:1**.
- Set the **Attack** box to **1.0** milliseconds.
- Set the **Release** box to **300.0** milliseconds.
- Click the **OK** button. Sound Forge compresses the file.

To set the RealAudio text fields:

- Click **Properties** on the File menu. The Properties window appears.

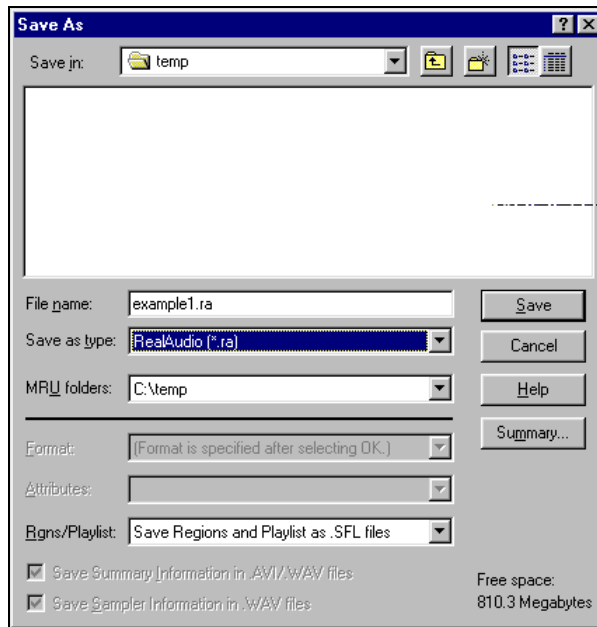
2. Click the **Summary** tab.



3. Enter the Title, Engineer, and Copyright information. This information will appear in the Title, Author, and Copyright text boxes on RealPlayer.
4. Click the **OK** button.

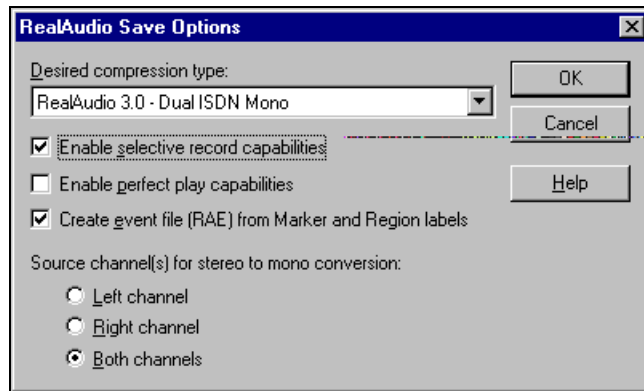
To save the file in RealAudio format:

1. Click **Save As** on the File menu. The Save As window appears.

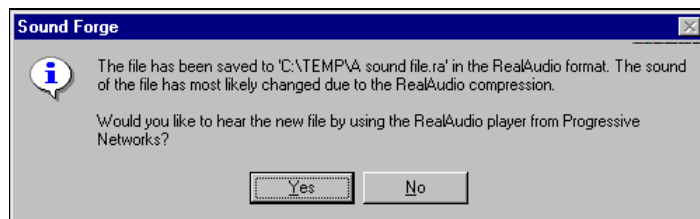


2. Enter the file name of the file you are saving.
3. Select **RealAudio (*.ra)** as the Save As Type.

- Click the **Save** button. The RealAudio Save Options window appears:



- Select the **Desired Compression Type**.
- If the file you recorded was stereo and you selected a mono Compression Type, select the **Stereo File Conversion**. It is recommend that you use the **Both Channels** option.
- Click the **OK** button. The following message appears:



- This message verifies that you have saved the file in RealAudio format. You can play the file by clicking the **Yes** button.

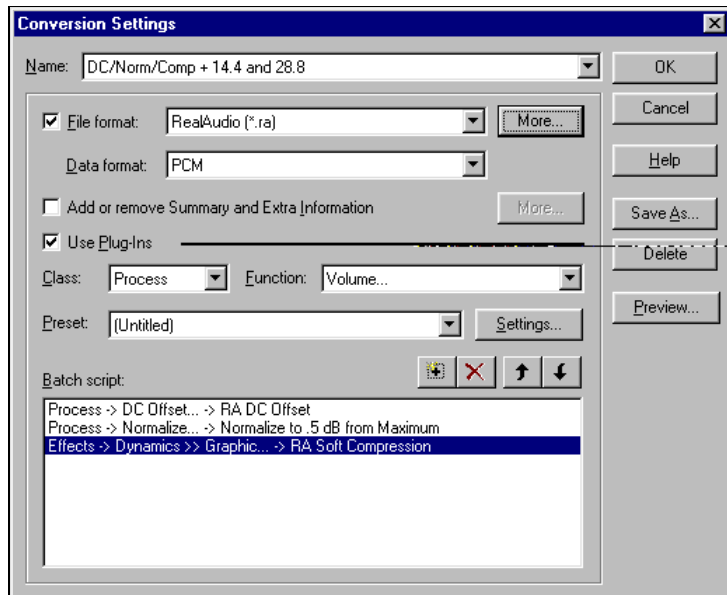
Batch Encoding with Sound Forge

Sound Forge provides a quick and easy tool for batch encoding. This tool is an add-on to the basic Sound Forge package. All the options and processes available in Sound Forge can be performed during a batch encoding. For instance, all the preprocessing described in “Encoding RealAudio files with Sound Forge” on page 234 can be saved as one conversion setting.

Note Progressive Networks has made available a script which incorporates all the recommended preprocessing for encoding RealAudio files at:
<http://www.real.com>

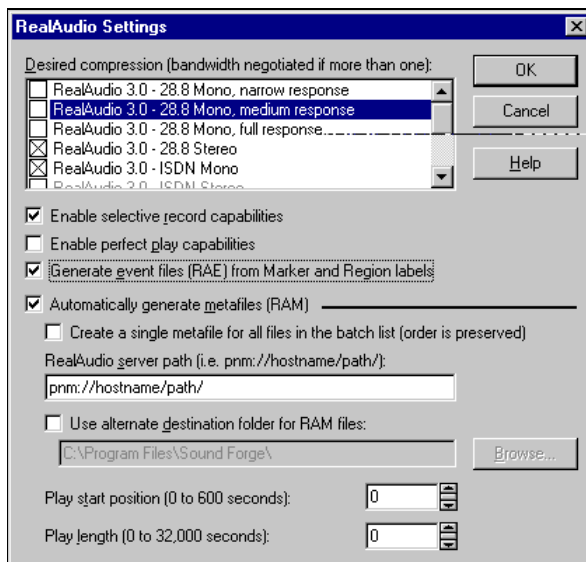
To batch encode with Sound Forge:

1. Click **Add** on the File menu.
2. Select the files you want to encode and add them to the list.
3. Click **Conversion** on the Options menu and select **Settings**.



4. Make a simple RealAudio bandwidth negotiation script, by doing the following:

- a) Check the **File Format** box and select **RealAudio (*.ra)** from the drop-down menu.
- b) Click the **More** button to access the RealAudio Settings window:



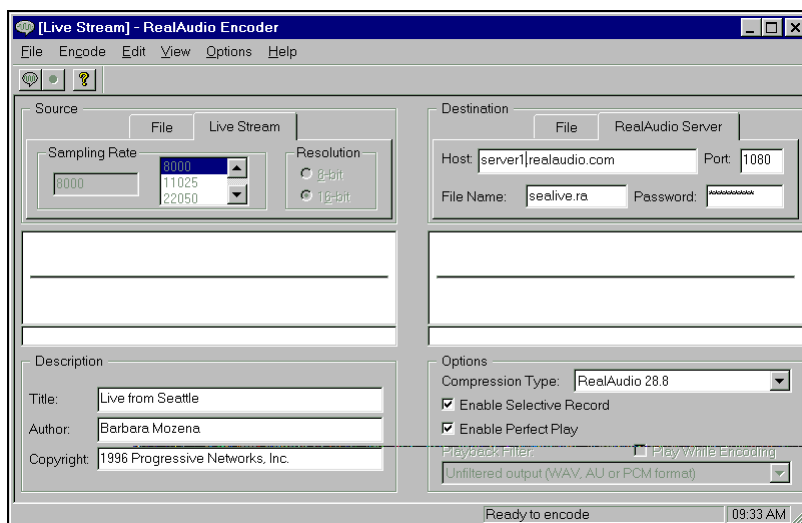
- c) Select one or more types of compression. If you select more than one, Sound Forge automatically sets them up for bandwidth negotiation. In addition, Sound Forge creates a .ram file if you check the **Automatically Generate Metafiles (RAM)** box.
 - d) Click the **OK** button in the RealAudio Settings window.
5. Click the **OK** button in the Conversion Settings window.
 6. Click the **Run Current Script** on the Convert menu. You have the option of having Sound Forge stop converting when it encounters an error or to continue encoding. Your files are encoded to your specifications.

Using RealAudio Encoder for Windows

RealAudio Encoder window is comprised of a menu bar, toolbar, encoding status display, and four information panes: Source, Destination, Description, and Compression.

There are two versions of Encoder for Windows: a free Encoder available for download from Progressive Networks web site and a Live Encoder that is included when you purchase RealServer. You can encode a static file using either Encoder.

Note RealAudio Encoder included with RealServer can deliver live content for broadcasting live events. For information about RealAudio Live Encoder for Windows, refer to “Live RealAudio Encoder for Windows” on page 296.



To encode a file using RealAudio Encoder for Windows:

1. In the Source pane, select the **File** tab to encode from a static file or select the **Live Stream** tab to encode from a live source. If you select the **Live Stream** tab, continue with Step 4.

Note The free RealAudio Encoder cannot broadcast a live event. If you have the RealAudio Live Encoder and want to broadcast a live event, refer to “Delivering Live Content” on page 261.

2. In the **File** tab, enter the directory and filename of the previously digitized audio file or click **Browse** to find the file. RealAudio Encoder automatically detects the sampling rate, format and resolution of a source file and display them in the **Format** box. If the Encoder cannot recognize these attributes, a dialog box appears where you can specify the information.

When you have selected the file to encode, a graphical representation of the file appears below the Source pane and the length of the clip is displayed. You can choose to not display the Audio Signal by clearing **Display Audio Signal** from the Options menu.

3. In the Description pane, enter the Title, Author, and Copyright information for your RealAudio output. These fields are optional.
4. If you want to allow RealPlayer Plus users to save your RealAudio signal to disk, select **Enable Selective Record**.
5. If you want to allow RealPlayer Plus users with slower connections (for example 14.4 Kbps modems) to experience RealAudio files encoded for a higher bandwidth by partially downloading audio data before beginning playback, select **Enable PerfectPlay**.
6. In the Destination pane, select **File** to encode a static file. If you have the RealAudio Live Encoder, and want to serve the file as you encode, select the **RealServer** tab. (This option is not available in the free RealAudio Encoder.)

Note For information about encoding live audio and the RealServer tab, refer to “Delivering Live Content” on page 261.


7. In the fields on the **File** tab, enter the name and location of the .ra output file. The Encoder suggests a default output file with the same name (but

with a .ra filename extension) and in the same directory as the source file. You can modify the output filename.

8. In the Compression pane, select the appropriate compression type.

For a description of each encoding algorithms, refer to “Choosing an Encoding Algorithm” on page 232.

9. If you want to listen to your audio as it is being encoded, select **Play While Encoding** from the Options Menu. Using this feature, however, may slow the encoding process.
10. If you select **Play While Encoding**, you can choose to listen either to the input file or to the decompressed .ra output file. Select one of these options from the Options menu.

11. Click , or the **Start Encoding** button to start the encoding process.

You may choose to view the progress indicator which shows the percentage completed or to view side-by-side plots of the input and output waveforms by clicking **Show Audio Signal** on the Options menu. Showing the waveforms may slow down the encoding process.

Note You can change Encoder default settings on the Preferences window. For more information, refer to “Setting Windows Encoder Options and Preferences” on page 246.

12. Listen to the newly encoded file with RealPlayer.

Setting Windows Encoder Options and Preferences

Windows Encoder enables to do set certain options and preferences. These options and preferences.

Show Audio Signal

RealAudio Encoder displays the audio signal for both the input and output files. Turning off this feature results in somewhat faster Encoding.

To change the setting of this option:

- Toggle **Show Audio Signal** from the Options menu.

Play While Encoding

You can listen to the audio as it is encoded if your PC has sufficient performance.

Note This option is not available for live input streams.

To play while encoding:

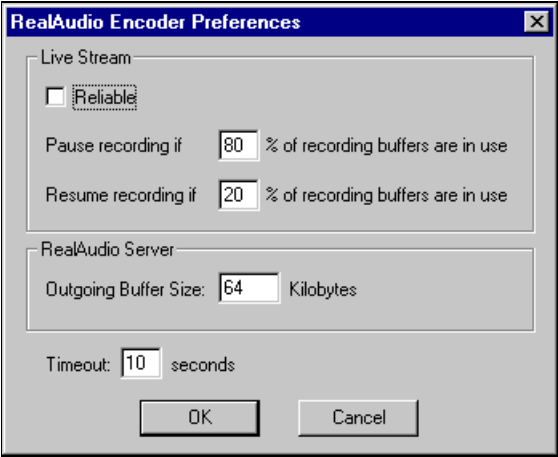
- Click **Play While Encoding** in the Options panel.

Note If the audio skips or is not available, your computer may not have sufficient performance to play correctly.

Preferences Window

To set preferences:

1. Click **Preferences** on the Options menu. The Preferences Window appears:



2. Set the appropriate settings.

Option	Description
Live Stream	<p>Click Reliable from a live source using a reliable connection that can keep up with the data flow. If this field is checked, and the recording buffers fill, you receive an error message and RealAudio Encoder stops encoding.</p> <p>If your connection is not reliable, set the other two fields. The Pause Recording field indicates the highest percentage to allow the recording buffers to reach. The Resume Recording field indicates at what percentage to begin recording after pause.</p>
RealServer	<p>This field is unavailable in the free Encoder.</p> <p>For the Live Encoder, this field indicates the capacity of the RealServer receiving the live broadcast.</p>
Timeout	<p>This field indicates when the Encoder should timeout if it cannot connect to RealServer.</p>

Set Default Settings

You can set certain Encoder options to default to a particular setting. These defaults appear in the Encoder window when you encode a file, and are applied to any file being encoded through drag-and-drop encoding.

Note When you are encoding a file in the Encoder window, you can modify the default settings on a file-by-file basis.

To set default Encoder options:


- Set the options on the Encoder window and click **Save Default Setting** on the Options menu.

The following options can be set through Save Default Setting:

- Title
- Author
- Copyright
- Compression
- PerfectPlay
- Selective Record

Drag-and-Drop Encoding for Windows

RealAudio Encoder for Windows supports two types of drag-and-drop encoding:

- Clicking an input-audio-file icon and dragging it onto an open Encoder window enters path and filename information into the appropriate Encoder fields. Then, you only need to enter the descriptive information and settings and click  to begin the encoding process.
- You can also drag and drop an input-file icon onto the RealAudio Encoder icon. The file will be encoded using the default output filename and the most-recently saved Description and Options. Be sure that you have saved the settings that you want applied. Refer to “Setting Windows Encoder Preferences” on page 246.

Command Line Encoding for Windows

In some situations you may find it convenient to encode within the DOS command line environment by creating a batch file to automate the encoding of many separate input files.

You can specify command line arguments that direct the Encoder to process an input file and then shut down. The syntax is:

```
ENCODER file_to_encode [settings_file]
```

where `file_to_encode` is the digital audio input file and `settings_file` is an optional file in which you have saved your encoding preferences.

For example, the command:

```
ENCODER one.wav prefs.txt
```

encodes the file `one.wav` using the Encoder settings specified in the file `prefs.txt`.

If you place the files you want to encode in the same directory as RealAudio Encoder, you do not have to specify file paths. However, if you want to encode files in other directories, you need to specify the complete paths to these files. Likewise, if your working directory is not the one in which your Encoder is installed, you need to specify its location. For example, the command:

```
C:\raencode\encoder C:\docs\one.wav C:\prefs.txt
```

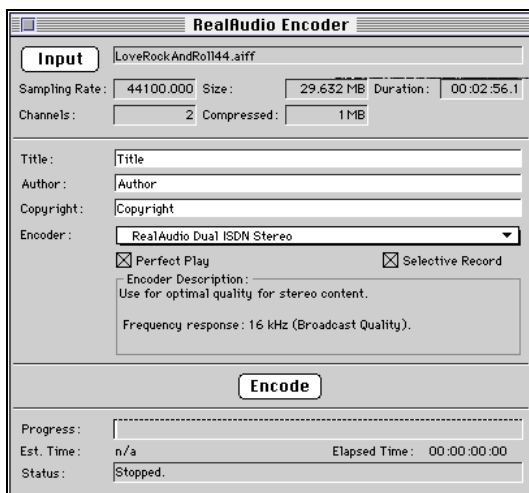
creates `one.ra` in the directory `C:\docs` from within any working directory.

Using the Encoder for Macintosh

There are two version of RealAudio Encoder for Macintosh: a free Encoder available for download from Progressive Networks web site and a Live Encoder that is included when you purchase RealServer. You can encode a static file using either Encoder.

Note RealAudio Encoder included with RealServer can deliver live content for broadcasting live events. For information about RealAudio Live Encoder

for Macintosh, refer to “Live RealAudio Encoding for Macintosh” on page 298.



To encode a file with RealAudio Encoder for Macintosh:

1. Select a previously existing audio file to convert into the RealAudio format, by clicking **Input**.
2. Select the file you want to compress from the Directory window and click **Open**. When you open a file, the Sampling Rate, Duration, Size, and Compressed Size appear at the top of the window.
3. Enter Title, Author, and Copyright information for your RealAudio output. These fields appear both within the main Encoder window and within the Preferences window. If you do not want to establish default values for this information, enter the information in the Encoder window.

Note If you plan to encode multiple files with the same information, you can save time by using the Preferences window to specify information that is common to all of your files, such as Copyright. Access the RealAudio Preferences window by choosing **Preferences** from the View menu. Refer to “Macintosh Preferences Window” on page 252.

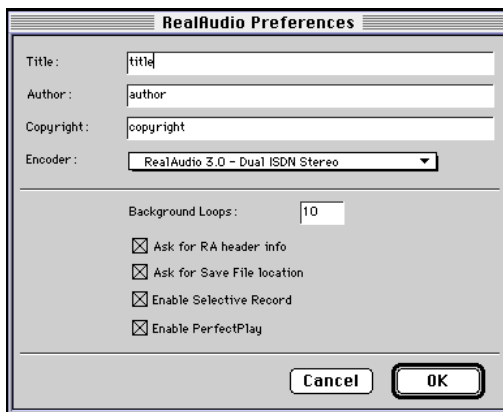
4. In the **Encoder** box, choose the encoding algorithm for this file. A description of the chosen algorithm appears in the Encoder Description field.

For a description of each encoding algorithms, refer to “Choosing an Encoding Algorithm” on page 232.

5. If you want to allow RealPlayer Plus users to save your RealAudio signal to disk, select **Enable Selective Record**.
6. If you want to allow RealPlayer Plus users with slower connections (for example 14.4 Kbps modems) to experience RealAudio files encoded for a higher bandwidth by partially downloading audio data before beginning playback, select **Enable PerfectPlay**.
7. Click the **Encode** button to start the encoding process.
8. Select folder and filename for the output file in the Directory window. While the file is encoding, the Progress bar at the bottom of RealAudio Encoder window indicates what percentage of the file is encoded.
9. Verify the encoding by playing the newly created file with a RealPlayer.

Macintosh Preferences Window

The Preferences window displays the default settings used by RealAudio Encoder. Information that you enter here is saved when you close this window. Do not enter information into the Preferences window unless you intend to set default parameters for encoding.



Feature	Description
Title	Sets the default Title.
Author	Sets the default Author.
Copyright	Sets the default Copyright.
Encoder	Sets the default Encoder algorithm.
Background Loops	<p>The number in this field controls how the Encoder shares CPU time with other active applications. The default is set to 10 for Power PC Macintoshes. The higher this value is set, the faster files encode. However, your computer will be less responsive if you switch to other applications.</p> <p>Note Macintosh 680x0 users may want to set this field to 1.</p>
Ask for RA Header Info	Select this check box to have a window appear before each file encodes. This gives you the opportunity to specify new Title, Author, and Copyright information for each file.
Ask for Save File Location	<p>Select this check box to have a Save File window appear as each file is encoded so you can specify the folder and filename for the RealAudio output file.</p> <p>If you do not select this check box and are encoding multiple files, the RealAudio files are saved in the same folder as the input file, with the .ra filename extension added.</p>
Enable Selective Record	Select this check box to allow RealPlayer Plus users to save your RealAudio clip to disk.
Enable PerfectPlay	Select this check box to allow RealPlayer Plus users with slower connections (for example 14.4 Kbps modems) to play RealAudio files encoded for a higher bandwidth by partially downloading audio data before beginning playback.

Drag-and-Drop Encoding for Macintosh

RealAudio Encoder for Macintosh supports drag-and-drop encoding. You can drag and drop input file icons onto the **RealAudio Encoder** icon. This automatically encodes files using the default output filename and the most recently saved encoding preferences. If you have selected the **Ask for RA Header Info** check box or the **Ask for Save File Location** check box, you are prompted for information as each file is encoded.

Encoding with AppleScripts for Macintosh

If you plan to encode large numbers of files on a regular basis, you may want to consider writing AppleScripts to perform batch encoding. This automatically encodes files using the default output filename and the most-recently saved encoding preferences. Sample AppleScripts are provided in the AppleScript Examples folder in the RealAudio Encoder folder. Three scripts are provided: one to encode a single file, one to encode multiple files, and one to record an Encoder session. A sample of an AppleScript is included in this document, refer “AppleScript Sample Code” on page 213.

The following is a list of AppleScript verbs:

Required Suite: Terms that every application should support

Verb	Description
open	Open the specified object(s).
print	Print the specified object(s),
quit	Quit application.
Run	Sent to an application when it is double-clicked.

Standard Suite: Common terms for most applications

Verb	Description
close	Closes an element.
Delete	Deletes an element.

Verb	Description
Get	Get the data for an object.
Make	Make a new element.
Set	Set an object's data.
encode	Encodes inFile into outfile in RealAudio format.
Class application	An application program.
Elements	Encoder by name.
Class encoder	<p>A RealAudio encoder object.</p> <p>Properties:</p> <p>Client Object reference—The AppleEvent target for status messages</p> <p>name string [r/o]—The title of the session.</p> <p>Title string—Title field of RA header in output file.</p> <p>Author string—Author field of RA header in output file.</p> <p>Copyright string—Copyright field of RA header in output file.</p> <p>Infoprompt boolean—Prompt user for RA header info before encoding.</p> <p>Saveprompt boolean—Prompt user for output file before encoding.</p> <p>Loopcount integer—Number of encoding loops before yielding to another application.</p> <p>Infile string—Input file to RealAudio Encoder.</p> <p>Outfile string—Output file for RealAudio Encoder.</p> <p>Processing boolean [r/o] —Flag to indicate if encoder is processing (processing == TRUE).</p> <p>Compression string—Name of encoder to use for encoding.</p>

Verb	Description
Class RACut	<p>Copies the specified amount of audio into output file.</p> <p>Properties:</p> <p>cutfile string—Input file to RealAudio Encoder. outfile string—Output file for RealAudio Encoder. string—Start time. string—End time.</p>
Class RAPaste	<p>Concatenates all input files into the specified output file.</p> <p>Properties:</p> <p>a list of string—Input files. outfile string—Output file for RealAudio Encoder.</p>

AppleScript Sample Code

This is an AppleScript example that demonstrates how to create an encoder session for RealAudio Encoder, as well as encoding the file.

```
tell application "RealAudio Encoder (PPC)"
    activate
    with timeout of 90000 seconds

    make encoder -- make a new encoder session
    copy result to myencoder -- save reference to the new
    -- encoder session

    -- set session default settings
    set infoprompt of myencoder to false -- don't ask for RA
    -- Info
    set saveprompt of myencoder to false -- don't put up
    -- Save File box

    -- select the encoder
    set compression of myencoder to "RealAudio Encoder 1.0"

    -- set processing loops
    set loopcount of myencoder to 10

    -- set Title, Author, and Copyright messages
    set title of myencoder to "Goodbye"
```



```
set author of myencoder to "My name"
set copyright of myencoder to "©1995 My Company Inc."

-- select input file
set infile of myencoder to ":AppleScript Examples:Sample
Sounds:Goodbye"
set outfile of myencoder to ":AppleScript xamples:Sample
Sounds:Goodbye.ra"

-- encode the file
encode myencoder

-- wait until processing is complete
repeat while myencoder is processing
end repeat

close myencoder
delete myencoder
quit

end timeout
end tell
```

Using RealAudio Encoder for UNIX

There are two versions of RealAudio Encoder for UNIX: a free Encoder available for download from the Progressive Networks Web site and a Live Encoder that is included when you purchase RealServer. You can encode a static file using either Encoder.

Note RealAudio Encoder included with RealServer can deliver live content for broadcasting live events. For information about RealAudio Live Encoder for UNIX, refer to “Live RealAudio Encoding for UNIX” on page 300.

RealAudio Encoder for UNIX is run from the command line, using the following syntax:

```
raencoder options file
```

where *options* are any of the RealAudio Encoder options described below and *file* is the input audio file to be compressed into the RealAudio format.

Note If no input filename is specified, STDIN is assumed.

By default, RealAudio Encoder for UNIX displays a progress indicator on your screen as the file is encoded. If you plan to encode large numbers of files on a regular schedule, you can write a shell script file for multiple encoding.

Option	Description								
-a	Use this option to specify an Author string.								
-c	Use this option to specify a Copyright string.								
-C	Use this option to specify the Codec with which to encode and decode this file. Valid options are: <div><table><tr><th>Codec</th><th>Algorithm</th></tr><tr><td>14_4</td><td>RealAudio 2.0 - 14.4</td></tr><tr><td>28_8</td><td>RealAudio 2.0 - 28.8</td></tr><tr><td>DNET</td><td>RealAudio 3.0 algorithms</td></tr></table></div> Default: 28_8	Codec	Algorithm	14_4	RealAudio 2.0 - 14.4	28_8	RealAudio 2.0 - 28.8	DNET	RealAudio 3.0 algorithms
Codec	Algorithm								
14_4	RealAudio 2.0 - 14.4								
28_8	RealAudio 2.0 - 28.8								
DNET	RealAudio 3.0 algorithms								

Option	Description																																	
-F	<p>Use this option to specify the Flavor of the Codec. Depending on the Codec chosen valid options are:</p> <table><thead><tr><th>Codec</th><th>Flavor</th><th>Algorithm</th></tr></thead><tbody><tr><td>14_4</td><td>0</td><td>RealAudio 2.0 - 14.4</td></tr><tr><td>28_8</td><td>0</td><td>RealAudio 2.0 - 28.8</td></tr><tr><td>DNET</td><td>0</td><td>RealAudio 3.0 - 28.8 Mono, narrow response</td></tr><tr><td>DNET</td><td>1</td><td>RealAudio 3.0 - 28.8 Mono, medium response</td></tr><tr><td>DNET</td><td>2</td><td>RealAudio 3.0 - 28.8 Mono, full response</td></tr><tr><td>DNET</td><td>3</td><td>RealAudio 3.0 - 28.8 Stereo</td></tr><tr><td>DNET</td><td>4</td><td>RealAudio 3.0 - ISDN Mono</td></tr><tr><td>DNET</td><td>5</td><td>RealAudio 3.0 - ISDN Stereo</td></tr><tr><td>DNET</td><td>6</td><td>RealAudio 3.0 - Dual ISDN Mono</td></tr><tr><td>DNET</td><td>7</td><td>RealAudio 3.0 - Dual ISDN Stereo</td></tr></tbody></table> <p>Default: 0</p>	Codec	Flavor	Algorithm	14_4	0	RealAudio 2.0 - 14.4	28_8	0	RealAudio 2.0 - 28.8	DNET	0	RealAudio 3.0 - 28.8 Mono, narrow response	DNET	1	RealAudio 3.0 - 28.8 Mono, medium response	DNET	2	RealAudio 3.0 - 28.8 Mono, full response	DNET	3	RealAudio 3.0 - 28.8 Stereo	DNET	4	RealAudio 3.0 - ISDN Mono	DNET	5	RealAudio 3.0 - ISDN Stereo	DNET	6	RealAudio 3.0 - Dual ISDN Mono	DNET	7	RealAudio 3.0 - Dual ISDN Stereo
Codec	Flavor	Algorithm																																
14_4	0	RealAudio 2.0 - 14.4																																
28_8	0	RealAudio 2.0 - 28.8																																
DNET	0	RealAudio 3.0 - 28.8 Mono, narrow response																																
DNET	1	RealAudio 3.0 - 28.8 Mono, medium response																																
DNET	2	RealAudio 3.0 - 28.8 Mono, full response																																
DNET	3	RealAudio 3.0 - 28.8 Stereo																																
DNET	4	RealAudio 3.0 - ISDN Mono																																
DNET	5	RealAudio 3.0 - ISDN Stereo																																
DNET	6	RealAudio 3.0 - Dual ISDN Mono																																
DNET	7	RealAudio 3.0 - Dual ISDN Stereo																																
-h	Use this option to display this list of options.																																	
-i	<p>Use this option to display the information stored in the header of your input file. For example, typing:</p> <pre>raencoder -i file</pre> <p>might produce the output:</p> <pre>WAVE Header Encoding is linear; 2 bytes per sample Number of bytes of audio is 23038. Sampling rate: 8000 # of channels: 1</pre>																																	
-l	<p>Required for live recording. Use to indicate source of audio plugged into your sound card. Valid options are:</p> <pre>line cd mic</pre>																																	

Option	Description				
-o	Use this option to specify the output RealAudio filename.				
-p	<p>Use this option to enable or disable PerfectPlay. In PerfectPlay mode, RealPlayer Plus uses an expanded audio buffer so users with slow connections (for example 14.4 Kbps modems) can play files encoded for higher bandwidths. Valid options are:</p> <table><tr><td>0</td><td>Disable PerfectPlay</td></tr><tr><td>1</td><td>Enable PerfectPlay</td></tr></table> <p>Default: 1</p>	0	Disable PerfectPlay	1	Enable PerfectPlay
0	Disable PerfectPlay				
1	Enable PerfectPlay				
-q	Use this option to disable the status indicator while encoding.				
-r	<p>Use this option to set the input file sampling rate. This option overrides the sampling rate specified in the header of the input audio file. Valid values are 8, 11, 16, 22, 32, 44.</p> <p>Default: 8</p>				
-s	<p>Use this option to specify the sample width of .pcm input audio data. Valid options are:</p> <table><tr><td>8</td><td>8-bit .pcm data</td></tr><tr><td>16</td><td>16-bit .pcm data</td></tr></table> <p>Default: 16</p>	8	8-bit .pcm data	16	16-bit .pcm data
8	8-bit .pcm data				
16	16-bit .pcm data				
-t	Use this option to specify a Title string. The string is displayed in the RealPlayer window.				
-v	Use to display the version number of RealAudio Encoder.				

Option	Description
-w	<p>Use this option to enable or disable Selective Record. Enabling Selective Record allows listeners to save your RealAudio data. Valid options are:</p> <p>0 Disable Selective Record 1 Enable Selective Record</p> <p>Default: 0</p>

The following are examples of the command line:

Encoding a .wav file with the RealAudio 3.0 - 28.8 Mono, medium response algorithm:

```
raencoder -t"My Title" -a"My Name" -c"My
Company, 1996" -omyfile.ra myfile.wav
```

Encoding an .au file with the RealAudio 2.0 - 14.4 algorithm

```
raencoder -fl -t"My Title" -a"My Name" -c"My
Company, 1996" -omyfile.ra myfile.au
```

RealAudio Batch Encoding

You can batch encode multiple files in several ways depending on your operating system and the tools you have available. For more information refer to the following sections:

- “Batch Encoding with Sound Forge” on page 242.
- “Encoding with AppleScripts for Macintosh” on page 254.
- “Batch Encoding” on page 241.

Editing Audio Files

If you want to edit your sound files, you have two options—you can edit your .ra files, or you can edit your original source files.

Editing Input Files

This is the editing method of choice. Encoded Audio is stored in indivisible clips of varying duration depending on the encoding algorithm. This limits how precisely encoded audio can be cut and pasted. Sound editing software can cleanly and easily cut and paste .wav or .au files, and there are numerous programs that provide such features as mixing (adding background music to a vocal segment) and fading in or out.

Editing RealAudio Files

RealAudio Encoder is distributed with two editing tools utilities, Racut and Rapaste, which enable simple editing of RealAudio files. The Racut tool cuts specific portions of a RealAudio file. The Rapaste tool combines two or more RealAudio files (encoded with the same algorithm) into one file.

Encoded Audio is stored in indivisible clips of varying duration depending on the encoding algorithm. This limits how precisely encoded audio can be cut and pasted. This is a compelling reason for editing the original source files rather than RealAudio files. The following table lists the indivisible duration for each encoding algorithm:

Encoding Algorithm	Duration
RealAudio 2.0 - 14.4	.02 seconds
RealAudio 2.0 - 28.8	1.4 seconds

Encoding Algorithm	Duration
RealAudio 3.0 - 28.8 Mono, narrow response	.139 seconds
RealAudio 3.0 - 28.8 Mono, medium response	.139 seconds
RealAudio 3.0 - 28.8 Mono, full response	.192 seconds
RealAudio 3.0 - 28.8 Stereo	.192 seconds
RealAudio 3.0 - ISDN Mono	.070 seconds
RealAudio 3.0 - ISDN Stereo	.096 seconds
RealAudio 3.0 - Dual ISDN Mono	.035 seconds
RealAudio 3.0 - Dual ISDN Stereo	.048 seconds

Editing RealAudio Files in UNIX and Windows

The RealAudio file editing tools, RACut and RAPaste, are run from a UNIX or DOS command line. An additional command line tool, Rax, is provided for editing header information of existing RealAudio files on these platforms. The rax tool can be used to change Title, Author, or Copyright strings or to modify Selective Record or PerfectPlay preferences.

Cutting RealAudio Files

Use the RACut tool to cut a specific portion of a RealAudio file.

To create a copy of a piece of a RealAudio file:

1. Listen to your original RealAudio file to determine the start and end times of the segment you want to copy (you can read the times on the status bar of your RealPlayer).
2. Create a copy by typing:


```
racut output.ra input.ra StartTime {+}EndTime
```

 where:

`input.ra` is the original file from which to copy
`output.ra` will contain the copied segment
`StartTime` is time into original file that the segment begins
`EndTime` is time into original file that the segment ends

Note A + (plus) in front of the `EndTime` indicates that the indicated time is the duration of the output clip rather than the time the output clip should end. For example: A `StartTime` of 2:00 and an `EndTime` of 3:00 results in a clip that is one minute long. A `StartTime` of 2:00 and an `EndTime` of +3:00 results in a clip that is three minutes long, starting two minutes into the original clip.

Both `StartTime` and `EndTime` follow the format:

[days:][hours:][minutes:]seconds[.tenths]

Bracketed data is optional—only seconds are required. However, to specify hour you must also specify minutes. To specify days you must give both hours and minutes.

The `StartTime` and `EndTime` must be an even multiple of the indivisible duration of each encoding algorithm. Otherwise, `Racut` rounds down to the closest such time.

Combining RealAudio Files

`Rapaste` creates a new RealAudio file that combines two or more existing RealAudio files that were encoded with the same RealAudio algorithm. The command uses the syntax:

```
rapaste output.ra in1.ra in2.ra [... inN.ra]
```

where:

`output.ra` is the name of the file you are creating
`in1.ra` is the input that comes first in the new file
`in2.ra` has content that will be second in the new file

At least two input files are required. The maximum possible number of input files depends on your command line limits.

The following examples demonstrate the use of the `Racut` and `Rapaste` tools.

1. Create a copy of the content beginning at the sixth hour of the first day and ending at the fourth hour of the third day, by typing:

```
racut output.ra input.ra 00:06:00 2:04:00:00.0
```

2. Create a file that contains the middle twenty seconds of a sixty-second clip, by typing:

```
racut excerpt.ra original.ra 20.0 40.0
```

3. Remove twenty seconds from the middle of a one-minute clip, by typing:

```
racut part1.ra original.ra 0 20.0  
racut part2.ra original.ra 40.0 01:00  
rapaste final.ra part1.ra part2.ra
```

4. Insert a clip at the 10-minute mark of an existing file, by typing:

```
racut begin.ra original.ra 0 10:00  
racut end.ra original.ra 10:00 endTime  
rapaste new.ra begin.ra insert.ra end.ra
```

Editing RealAudio Files on Macintosh

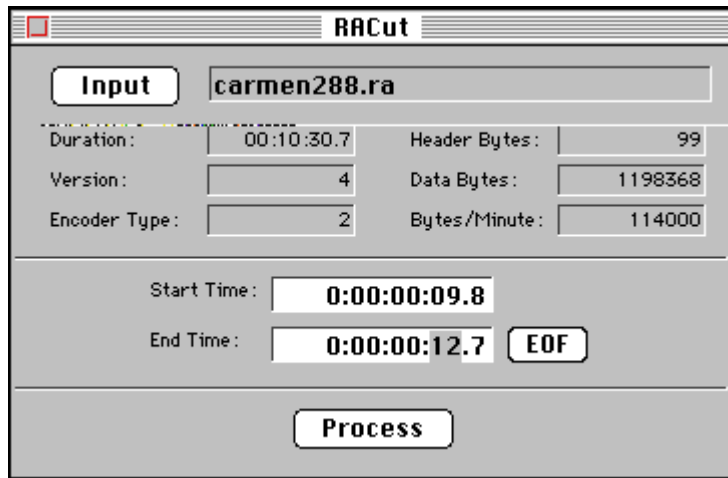
Copying Selected Portions of RealAudio Files

Use RACut to create a copy of a selected portion of a RealAudio file.

To create a copy of a selected portion of a RealAudio file:

1. Listen to the file and use the time display of your RealPlayer to determine the start and end times of the segment you want to capture.

2. Click **RACut** on the Process menu of RealAudio Encoder.



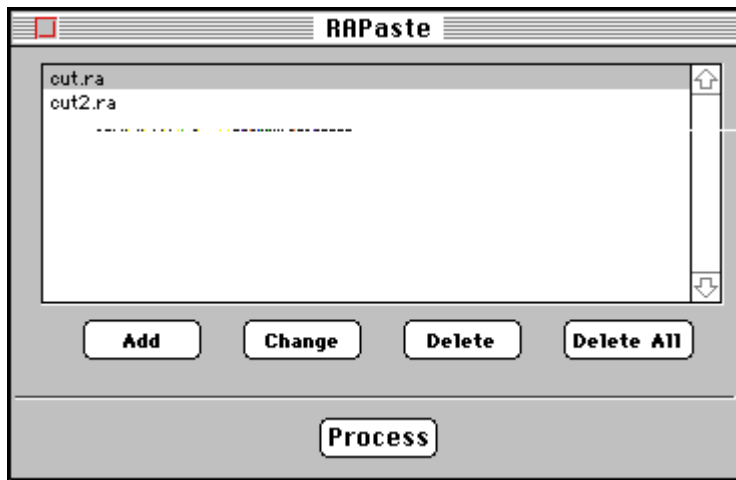
3. Click the **Input** button to browse for the .ra file from which you want to copy a segment.
4. Select your source file in the Directory window and click the **Open** button. Once you have opened the file, the RACut window displays its header information.
5. Type the limit times for the segment you want in the **Start Time** and **End Time** text boxes. If the segment you want extends to the end of the file, click the **EOF** button.
6. Click the **Process** button; you are prompted for the name of the new file. Type the new name and click the **Save** button.

Combining RealAudio Files

RAPaste creates a new RealAudio file that is a compilation of two or more existing RealAudio files.

To paste together multiple RealAudio files:

1. Click the **RAPaste** button on the Process menu of RealAudio Encoder



2. Click the **Add** button to select .ra files you would like to combine. Each file you select is added to the Paste list in the RAPaste window.

Click **Change** to modify a particular file in the Paste list. Click **Delete** to remove a file from the Paste list. Click **Delete All** to remove all of the files from the Paste list.

3. When the list of files is complete, click **Process**. You are prompted for the name of the new .ra file. Enter the name and click **Save**.
4. Use RealPlayer to verify the content of the new file.

Modifying RealAudio File Descriptions

You can change the Title, Author, or Copyright text, and modify the Selective Record and PerfectPlay settings in a .ra file using several different methods:

Method	Description
RealAudio Encoder	<p>Open a .ra file, edit the text fields, and click Encode. RealAudio Encoder modifies the fields. RealAudio Encoder does not re-encode the file, it just modifies the text strings.</p> <p>Note Turn off the Listen While Encoding option, otherwise, the entire file will be played.</p> <p>Refer to “Encoding RealAudio Clips” on page 229.</p>
Rax Tool	<p>This is a command line tool available for Windows and UNIX.</p> <p>Refer to “Rax Tool” on page 269.</p>
.ram File	<p>Set the Title, Author, and Copyright text strings in the .ram file.</p> <p>Refer to “Metafiles” on page TBD.</p>
Events Compiler	<p>Set the Title, Author, and Copyright text strings in the .rae file using either of these tools.</p> <p>Refer to “Cevents” on page 165.</p>

Rax Tool

Using the Rax tool you can modify the text strings or setting of a .ra file. The Rax tool is run from the command line by typing:

```
rax [options] file |dir
```

where the following options are available:

Feature	Description
-a	Use this option to specify an Author string.
-c	Use this option to specify a Copyright string.
-f	Use this option to fix a corrupted RealAudio file header.
-help	Use this option to print the options list to the screen.
-i	Use this option to display the .ra file header information.
-o	Use this option to specify the output RealAudio file name. If you do not use this option, then the input file is overwritten by the updated file.
-p	Use this option to enable or disable PerfectPlay. When this option is enabled, RealPlayer Plus users with slow connections (for example 14.4 Kbps modems) to experience RealAudio files encoded for a higher bandwidth by partially downloading audio data before beginning playback. Valid options are: 0 Disabled 1 Enabled
-t	Use this option to specify a Title string.
-v	Use this option to display the version number of the RealAudio Exchange tool.
-w	Use this option to enable or disable Selective Record When this option is enabled, RealPlayer Plus users can save your RealAudio signal to disk. Valid option are: 0 Disabled 1 Enabled

Sample Rax Commands

To change the title text in a .ra file:

- Type the command:
`rax -t "My New Title" file.ra`

To change the title of all the .ra files in the current directory:

- Type the command:
`rax -t "My New Title" *.ra`

To create a new file and not overwrite the original .ra file:

- Type the command:
`rax -t "My New Title" -ofile_new_title.ra file.ra`

To display .ra file header information:

- Type the command:
`rax -i file.ra`

To display .ra file header information for all .ra files in a directory:

- Type the command:
`rax -i rfile_dir`

Bandwidth Negotiation

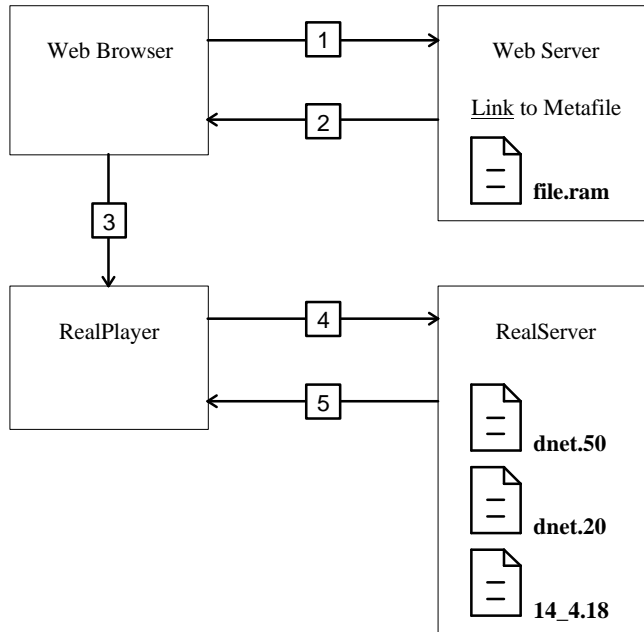
You can configure RealServer to deliver files encoded with different algorithms based on the capability of the user's RealPlayer. The Player gets the best quality the connection can handle without having to explicitly choose among multiple links and Web pages. You can choose to provide as many versions of each file as you want. The bandwidth negotiation process is transparent to users.

Bandwidth negotiation requires only one link on your Web site to a particular clip. To provide content in multiple formats without bandwidth negotiation, your Web site must have a separate hypertext link and metafile for each format.

For information on using Bandwidth Negotiation with Synchronized Multimedia presentations, refer to "Using Synchronized Multimedia with Bandwidth Negotiation" on page TBD.

The following diagram shows how RealPlayer and RealServer determine which file to play based on bandwidth.

File organization is the key to bandwidth negotiation. This figure shows a single link on a Web page, and three available RealAudio files encoded using different algorithms.



The following steps correspond to the numbers in the figure:

1. The user clicks a link to a RealAudio metafile on a Web page.
2. The Web server returns the metafile to the Web browser and based on the **.ram** file extension, sets the MIME type of the metafile to **audio/x-pn-realaudio**.
3. The Web browser looks up the MIME type of the metafile, starts RealPlayer as a helper application, and passes it the metafile.
4. RealPlayer reads the first URL from the metafile and requests it from RealServer. Based on its preference settings, RealPlayer also sends a list of RealAudio compression types it supports.
5. RealServer checks the directory specified by the URL and begins streaming the highest bandwidth file supported by RealPlayer.

The name of the RealAudio file specified in the URL in the metafile is actually a directory on the RealServer computer with the .ra filename extension. Within that directory are the individual files for each format. Name the files based on the following table, which is sorted in order of increasing bandwidth:

Encoding Algorithm	Filename
RealAudio 2.0 - 14.4	14_4.18
RealAudio 3.0 - 28.8 Mono, narrow response RealAudio 3.0 - 28.8 Mono, medium response RealAudio 3.0 - 28.8 Mono, full response	dnet.20
RealAudio 3.0 - 28.8 Stereo	dnet.25
RealAudio 2.0 - 28.8	28_8.36
RealAudio 3.0 - ISDN Mono RealAudio 3.0 - ISDN Stereo	dnet.50
RealAudio 3.0 - Dual ISDN Mono RealAudio 3.0 - Dual ISDN Stereo	dnet.100

Note that several encoding algorithms have the same filename. That means that you can deliver only one of those formats for any given URL.

The following table shows the file formats that each version of RealPlayer requests and in what order they are requested:

Player Version	Bandwidth Setting	Files Requested
1.0	All	14_4.18
2.0	14.4	14_4.18
2.0	28.8, ISDN, T1	28_8.36 14_4.18
2.1	14.4	14_4.18
2.1	28.8, ISDN, T1	28_8.36 14_4.18

Player Version	Bandwidth Setting	Files Requested
3.0	14.4	14_4.18
3.0	28.8	28_8.36 dnet.25 dnet.20 14_4.18
3.0	ISDN	dnet.50 28_8.36 dnet.25 dnet.20 14_4.18
3.0	T1	dnet.100 dnet.50 28_8.36 dnet.25 dnet.20 14_4.18

Note RealAudio Player 3.0 and later with a 28.8 Kbps connection always plays the RealAudio 2.0 - 28.8 (28_8.36) format if it is available. If you want to provide another 28.8 format such as 28.8 Stereo (dnet.25), do not also provide the 28_8.36 format. If you do not provide any format supported by a Player, that Player receives a message to upgrade to the current Player release.

Bandwidth Negotiation Example

In this example, you deliver one of three RealAudio formats depending on the connection speed and RealPlayer version.

To setup this example:

1. Encode the source file in the following formats:
 - RealAudio 2.0 - 14.4
 - RealAudio 3.0 - 28.8 Stereo
 - RealAudio 3.0 - ISDN Stereo

2. Create a metafile named **mozart.ram** containing a URL such as:

```
pnm://audio.realaudio.com/music/mozart34.ra
```

3. Create a link to the metafile in a Web page. The following HTML code is a typical link:

```
<A HREF="http://www.real.com/cl/mozart.ram">  
Listen to Mozart</A>
```

4. On the RealServer computer, create a directory named **mozart34.ra** in the **/music** directory.
5. In this directory, store the three RealAudio files, renamed **14_4.18**, **dnet.25**, and **dnet.50** as shown in the previous table. You can do this manually, use the **raconv** utility described in the next section, or write your own automation script.

The file that is played depends on the Player connection and version:

- ISDN or faster connection with RealAudio Player 3.0 or later: RealAudio 3.0 - ISDN Stereo format (**dnet.50**)
- 28.8 Kbps connection with RealAudio Player 3.0 or later: RealAudio 3.0 - 28.8 Stereo format (**dnet.25**)
- 14.4 Kbps connection with RealAudio Player 3.0 or later: RealAudio 2.0 - 14.4 format (**14_4.18**)
- 14.4 Kbps or faster connection with RealAudio Player version 2.1 and earlier: RealAudio 2.0 - 14.4 format (**14_4.18**)

Note 1 If you do not supply a RealAudio 2.0 - 14.4 or RealAudio 3.0 - 28.8 format file, users with RealAudio Player 2.1 or earlier receive an error message that they need to upgrade their Player.

Note 2 If you supply a RealAudio 2.0 - 14.4 format file (**14_4.18**), RealAudio Player 3.0 or later with a 28.8 Kbps connection always plays the 28.8 format file; the 28.8 Stereo format file is never played.

Using the Bandwidth Negotiation Utility

The Raconv utility helps you arrange your files into the organization required for bandwidth negotiation by generating the directory with the .ra extension and placing the appropriately renamed files in that directory. The utility uses information in the RealAudio file to determine how to rename the file. Because the utility renames files, keep a back up of your original files until you are sure that the process was successful.

Note The Raconv utility does not convert between RealAudio formats. Use RealAudio Encoder to create a file with each needed format.

To organize your files for bandwidth negotiation:

1. Encode your RealAudio files in the formats you want to support.
2. Store your recorded files in separate directories, one for each final format name. For example, RealAudio 3.0 - 28.8 Mono, narrow response and RealAudio 3.0 - 28.8 Mono, medium response go in the same directory, because they are both renamed dnet.20. The files that contain the same audio source encoded in different formats must have the same name. For example, if the URL specifies mozart34.ra, you need file named **mozart34.ra** in each directory.
3. Type the command:

```
raconv <InputFileName> <ContentDirectory>
```

Where InputFileName is the file to be turned into a directory and underlying RealAudio file and ContentDirectory is the directory in which you want to create the content directories.

4. Repeat the command for each RealAudio format you encoded.

For example, typing the command:

```
raconv /28_836files/mozart34.ra /music
```

creates the directory **/music/mozart34.ra**, moves the file **mozart34.ra** from the **28_836files** directory to this new directory, and renames the file **28_8.36**.

If your files are organized by encoding format, you can run **raconv** on a whole directory by entering wildcards for `InputFileName` . For example, typing the command:

```
raconv /28_836files/*.ra /music
```

This command takes all the RealAudio files in the directory **28_836files** and creates new directories and files under the directory **/music**.

The **raconv** utility prompts you before overwriting existing files. Use the `-f` option to force overwriting without prompting.

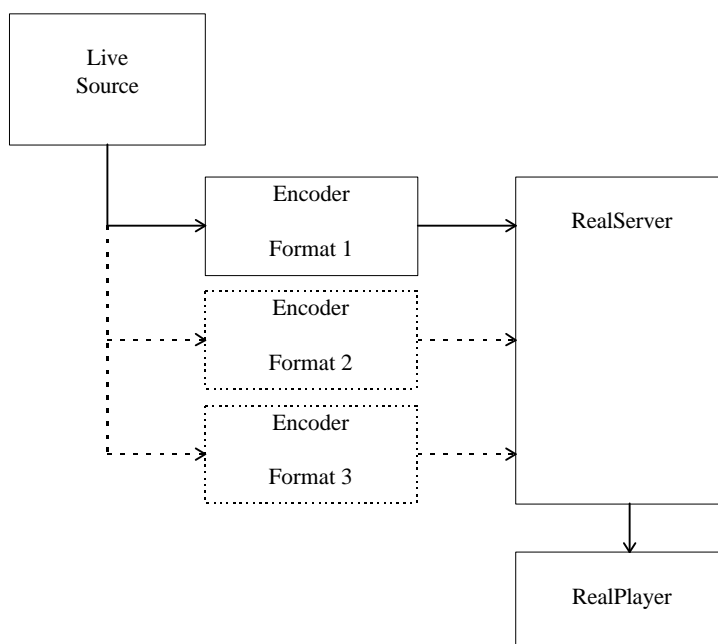
You can run **raconv** on files in one format to create the directories and then run **raconv** on files in other formats and place them in the appropriate directory.

Live Broadcasting

With RealServer, you can send live events directly to user's computers, letting people enjoy music, speeches, and public events from their computers. Whether you are promoting a concert, holding a company meeting, or covering a campaign speech, you can use RealAudio and RealVideo to attract whole new audiences.

Broadcasting live requires using RealServer with a RealAudio or RealVideo Encoder for transmission over the Internet. If you choose, you can save the event to disk while you are broadcasting.

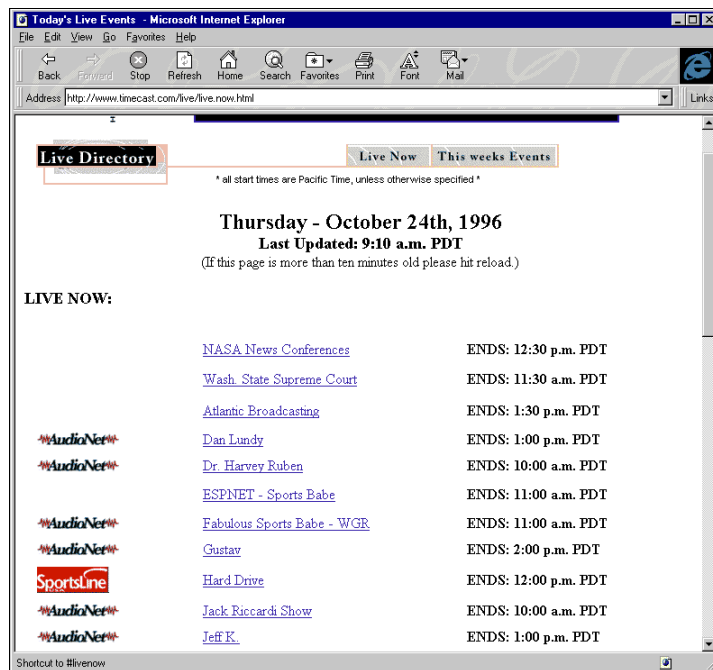
For each encoding algorithm that you want to broadcast live, you need an Encoder. For example, to offer a live broadcast in both RealAudio 2.0 - 28.8 and RealAudio 3.0 - 28.8 Stereo, you need to run two copies of RealAudio Encoder sending their output to RealServer. The bandwidth negotiation feature of RealServer automatically delivers the highest bandwidth signal supported by the Player.



Advertising Your Live Event

If you want to advertise your live event, you can list it in Timecast: Your RealMedia Guide (<http://www.timecast.com>), Progressive Networks online resource for RealAudio and RealVideo sites and live events. Thousands of people use Timecast to determine what broadcasts are available. By listing your live event on Timecast, you can greatly increase your audience.

Your listing appears in the Live Directory:



There are two ways to list your live events.

- Fill out a Web form with the relevant information.
- Place special tags in your HTML code.

Filling out a Web Form

Use the Web form (<http://www.real.com/live/addlive.html>) if you have only a few events or events/programs that occur at the same time each day, or once a week. On the Web form, you enter basic information about the event, such as name, complete URL, date and time of the event, and a short description. After submitting the form, the data is verified by a staff member of Timecast and posted on Timecast.

Place Special Tags in your HTML Code

If you have several different events, you can save time by inserting special “live tags” in your HTML code. The live tags are placed within comment lines in the source html, so that they are invisible to your users. A software robot reads the information in the live tags and automatically enters the event(s) in Timecast.

To advertise your live event using special tags:

1. On the web page with the link to your live event, insert the live tags using the following format:

```
<!--@Rastart event="Brief Event Description"
start="Thu, 11 Apr 1996 20:30:00 EST" -->
(some HTML that describes the event)
<!--@RAend-->
```

Note For more information about the live tag, refer to:
<http://www.real.com/help/content/livetags.html>

2. Send e-mail to: *live_event@prognnet.com*. Do not enter a subject. In the body of the e-mail, enter the complete URL for the page containing the live tags (for example: <http://www.real.com/example/tags.html>). A software Robot gathers the information in your live tags. The data is verified by a staff member of Timecast and posted.

Note It is VERY important that you include the full address, including the “http://” part. If you use frames, be sure to send the address for the event page that appears within a frame, not the HTML page that creates the frames.

Delivering Live Content

The RealAudio and RealVideo Encoders included with RealServer can deliver live content for broadcasting live events. The free RealAudio Encoder available from the Progressive Networks Web site does not support live delivery.

To deliver live content, you need:

- A live source
- A computer running a RealAudio or RealVideo Encoder
- A server running a RealServer

The computer running the encoder and the computer running RealServer can be on different platforms.

Setting Up RealServer

To enable live delivery, be sure the following configuration settings are included in the RealServer configuration file:

- **PnaPort** - the port number to which the Encoder connects
- **EncoderPassword** - the password the Encoder uses to connect

You have the option of specifying that the Server saves the live content as a RealAudio file by setting the **LiveFileTarget** and **LiveFilePassword** configuration option.

Connecting the Encoder to RealServer

The RealAudio and RealVideo Encoders translate the broadcast into one or more formats that RealServer can distribute over the Internet. As administrator

of the RealServer, you must provide a way for Encoder to connect to RealServer. You do this using the **PnaPort** and **EncoderPassword** configuration settings.

The **EncoderPassword** setting specifies the password the Encoder must use to connect to RealServer. Passwords are necessary to keep unauthorized users from connecting to the stream of your live broadcast. For example, if you have the setting:

```
EncoderPassword FrogNet
```

the person starting the Encoder must use the password `FrogNet` to start the Encoder.

Bandwidth negotiation during live events is a feature from RealAudio 3.0 that is no longer supported in RealServer. If you are using RealAudio 3.0, connect one encoder for each encoding algorithm you want to support. Specify the same file name as the output from each encoder. RealServer recognizes the format of each stream and directs it to RealPlayers requesting that format.

The **EncoderTimeout** configuration parameter specifies how long RealServer stays connected to an encoder that is not sending data.

EncoderPassword

Password used by RealAudio Encoder, RealVideo Encoder, and the slta utility program to connect to RealServer.

Used by	pnservice
Default value	(none)
Range of values	Alpha-numeric string without spaces
Restart Server after change	No

```
EncoderPassword <password>
```

Note that the default setting of no password allows any RealAudio Encoder or RealVideo Encoder to connect to the server.

Example

```
EncoderPassword rmRecord1
```

EncoderTimeout

The time in seconds that the Server will wait before disconnecting a RealAudio Encoder or RealVideo Encoder that is not sending data. If the connection to the Encoder is lost, the Server must disconnect before the Encoder can reconnect.

Used by	pnserver
Default value	30
Range of values	1 - 32767 seconds
Restart Server after change	No

```
EncoderTimeout <seconds>
```

Setting EncoderTimeout to less than 10 seconds is not recommended.

Example

```
EncoderTimeout 20
```

RealVideo Live Broadcasts

With the appropriate RealServer license key, RealVideo Encoder can deliver live content for broadcasting live events.

To deliver live content, you must:

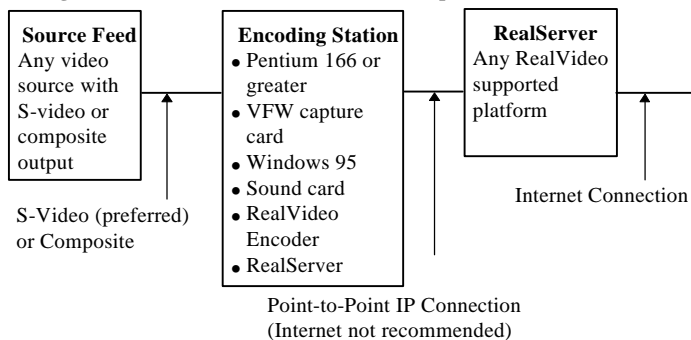
1. Install RealVideo encoder on the encoding machine.
2. Install RealServer on the serving machine. (It may be the same as the encoding machine.)
3. Attach your video source to the video capture card.
4. Set up your preferred audio device to your audio card (not the video card).

RealVideo Encoder and RealServer computers can be on different platforms.

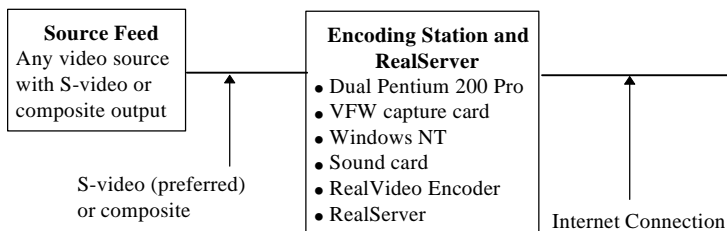
System Requirements

Delivering live content at a higher frame rate requires a different system configuration than delivering live content at a low frame rate. Three recommended configurations are detailed below.

Configuration 1: Low Frame Rate (<2 frames per second)

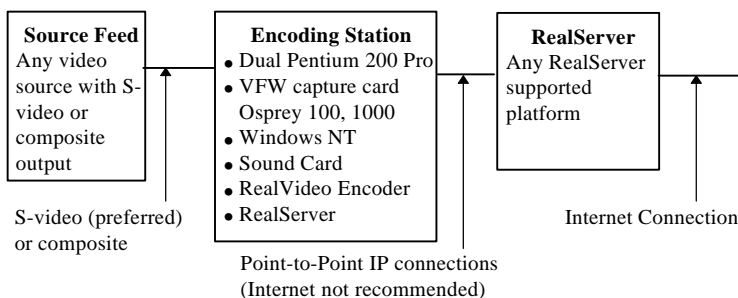


Configuration 2: High frame rate (up to 10 frames per second* <200 streams)



*Actual maximum frame rate is dependent on bit rate and content.

Configuration 3: High frame rate (up to 15 frames per second)



Software Requirements

To broadcast live, the following software is required:

- RealVideo Encoder
- A minimum of a 60 stream RealServer
- RealPlayer

Producing High Quality Live Content

- The average bit rate should be equal to the requested bit rate (the bit rate specified in the template). If the average bit rate exceeds the requested bit rate, you can:
 1. Switch to optimal frame rate.
 2. Use a lower bit rate audio codec.
 3. Use a higher requested bit rate.
- Audio quality is important when delivering live content. Setting the volume too low can result in a weak signal. Setting the volume too high can cause a distortion of the audio.
- Under no circumstance is loss of audio acceptable. If you experience audio loss in a live broadcast:
 1. Use optimal frame rate to ensure you don't send data at too high of a data rate.
 2. Test data with similar content before using a set frame rate.
 3. Lower the frame rate. If necessary, lower the video quality.
- Latency, the amount of time required for buffering the video, should not exceed 5-10 seconds when delivering live content.
- View the live content using a RealPlayer. This allows you to see how the live content looks when it is delivered.
- If you are not maintaining the required frame rate, reduce the system load by disabling the encoder preview window.

1. If you are having encoding problems you have five variables to adjust:
2. Frame size
3. Frame rate
4. Bit rate
5. Audio codec (portion of total bits)
6. Quality

You may need to experiment with these variables to achieve the overall effect you are after.

Encoding Templates

Live encoding usually requires creating your own template. How you design your template depends largely on the power of your system. When encoding video frames that are 176 pixels by 144 pixels, a dual processor 200 MHz computer gives a good result with a frame rate of 4 fps, Quality 100, or with a frame rate of 9 fps, Quality 25. A single processor Pentium system gives a better result when set at 2-3 fps.

For more information about creating templates, refer to “Creating Templates” on page 198.

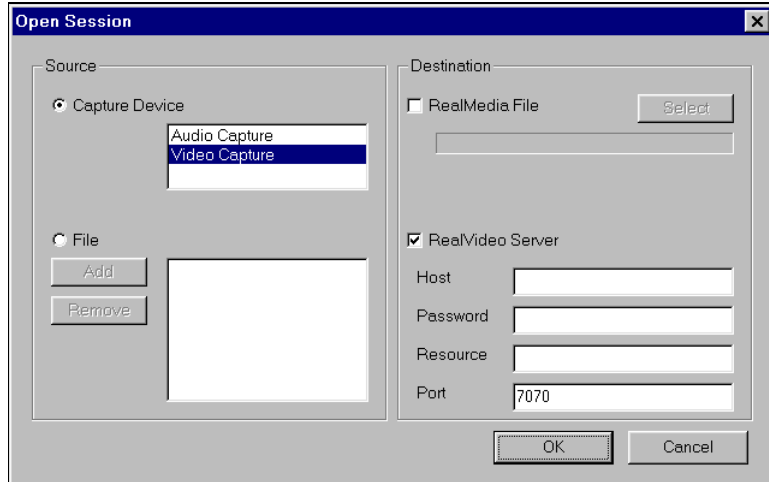
Live RealVideo Encoder for Windows

RealVideo Encoder for Windows runs on Windows 95 and Windows NT.

To provide content in multiple compression algorithms, run multiple machines using RealVideo Encoder with the same input signal. All concurrently running copies of RealVideo Encoder should use identical settings except for **Compression Type** and stream name.

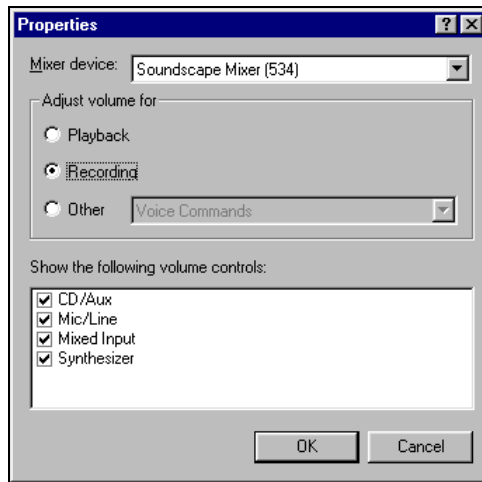
To encode and broadcast live content in Windows:

1. Start RealVideo Encoder. RealVideo Encoder window opens.
2. Click **Open Session** on the File menu. The Open Session window appears:

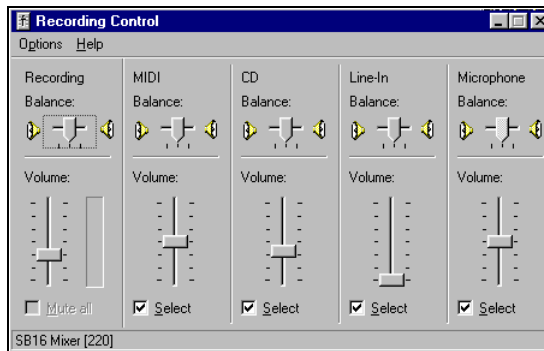


3. In the source pane, click the **Capture Device** button.
4. Click **Audio Capture** and **Video Capture**. Both audio and video must be captured together.
5. In the destination pane, click **RealServer** checkbox, a check appears in the box.
6. In the Host field, type the server name, IP address or machine name.
7. In the Password field, type the password. This field is optional.
8. In the Resource field, type the RealPlayer stream name.
9. Click the **OK** button. The Open Session window closes.
10. Select **Volume Control** from the Options menu. The Volume Control window displays.

11. Select **Properties** from the Options menu. The Properties window displays.

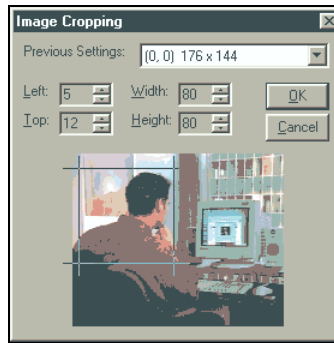


12. Click the **Recording** button to adjust the recording volume. (Playback adjusts the level of the volume you hear while you are encoding, not the volume the end-user hears.)
13. Click to place a check in the volume controls for the type of input you will be using and click the **OK** button. The Volume Control window displays with the volume controls you selected.



14. Click to place a check in the Select box for the type of input you will be using.

15. Adjust the sound level by moving the sliders up or down. Remember, if the volume is too high, the sound will be clipped. If the volume is too low, it will be difficult to hear. The best method for adjusting sound is to have a second machine nearby so that you can hear the result during broadcast.
16. Close the Recording Control window. The RealVideo Encoder window displays.
17. If you want to crop the image, select **Crop Input Image** from Options menu. The Image Cropping window displays. Notice the crop lines around the image. These lines show the portion of the image that will be encoded. Use the Left, Top, Width, and Height up and down arrows to adjust the size and location of the crop lines.



18. Close the Image Cropping window. The RealVideo Encoder window displays.
19. In the properties pane, enter the Title, Author, and Copyright information for your output stream. These fields are optional.
20. If you want to allow RealPlayer Plus users to save your RealVideo signal to disk, click the **Selective Record** checkbox.
21. In the templates pane, select the template to encode the file. For more information, refer to “Encoding Templates” on page 185.

Delivering live content usually requires that you create your own template. Refer to “Encoding Templates for Delivering Live Content” on page 288 and “Creating Templates” on page 198.

22. Select **Statistics** from the View menu. The Statistics window appears. The information displayed in this window during broadcast will help you evaluate and adjust your template settings.



24. Click the **Start** button to start the encoding process. The broadcast previews in the right window.

Note Preview can be turned off to marginally increase performance. From the Options menu, click **Show Filtered Output** to clear the selection.

25. Select **Set Video Format** from the Options menu. You can use this option to adjust video format settings while delivering live content. Click the **OK** button to accept the current settings.
26. In the results pane of the Statistics window, notice the **Frame Rate**. Wait while enough data is collected for an average frame rate to be determined. If the average frame rate is below the frame rate set in your template, adjust the encoding by varying one of the five variables.
27. In the results pane, notice Latency. If latency is greater than 5 or 10 seconds, adjust the template.
28. View the live content in the preview window. Viewing the live content through your Web browser is strongly recommended.

Live Encoding Command-Line Control

Command line options allow automation of live feeds. Refer to “Command Line Encoding for Windows” on page 195 for a list of static encoding options. Additional RealVideo Encoder options for live encoding are listed below:

Option	Description
/L	Turns on Live Input instead of static input file. The destination can then be selected between server or file output.
/S server[:port]/filename	Specifies delivery to a server on the supplied port. If the port is not supplied then the default of 7070 is used. The server can be either a hostname or IP address. The filename is used as the target on the server.
/W password	Specifies the password, if required to connect to the Live Server. The default is no password and is only used if the target is a server.
/O	Used for the file name at the encoder if simultaneous file output is required.
/I	Specifies an Input File.
/D hh:mm:ss	Allows the encoder to run for the supplied time. The maximum time is 999:59:59. When the time elapses then the encoder should gracefully shutdown and exit. If a static file is being encoded and the file ends before the maximum duration, there is no need for an error message. If this option is not supplied, then the encoder continues to run.

Ctrl-C terminates encoding.

Enhancing Live Broadcasts with Video Image Maps and Synchronized Web Pages

You can deliver a Synchronized Multimedia presentation or apply an image map to a video stream with a live broadcast. The events are delivered relative to the time a client begins playing the live broadcast, instead of relative to an absolute position in the broadcast.

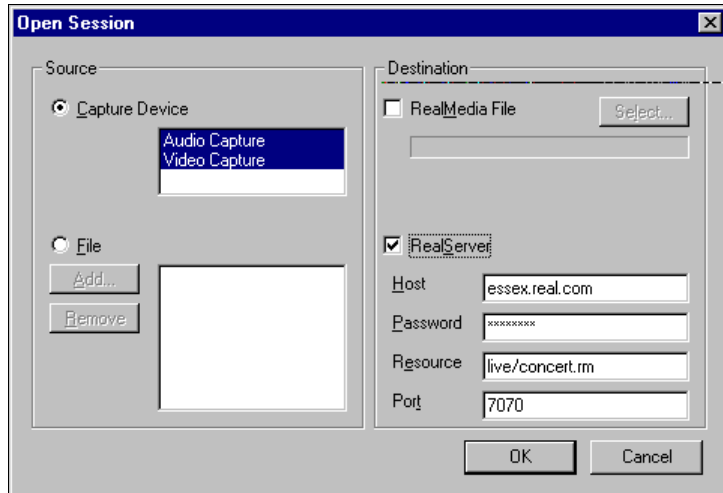
The URL events must be defined before the broadcast, but you can change the content of the Web page the URL references during the broadcast.

An image map can change over time, but this is not synchronized with the video stream. You can add any static content type to a live stream that is not already present in the stream. An example of this could be a link for the whole image to a site with a link to the broadcaster tied to a superimposed logo added to the stream before encoding.

To create a Synchronized Multimedia presentation for a live broadcast:

1. Create the input file that defines each event and the time that the event is sent. Remember that the times you specify are relative to when the client begins playing the live broadcast.
2. Run the **rmmerge** tool to create a .rm file. Repeat steps 1 and 2 for any image maps.
3. Name the .rm file with the same filename you specify in the **Resource** box in RealVideo Encoder for the broadcast.
4. Move the .rm file to the path specified in the **Resource** box in RealVideo Encoder.

For example, if you set up RealVideo Encoder as follows:



The file must be named **concert.rm** and it must be located in the **/live** directory relative to the server's base path.

RealAudio Live Broadcasts

With the appropriate license key, RealAudio Encoder can deliver live content for broadcasting live events.

To deliver live content, you must:

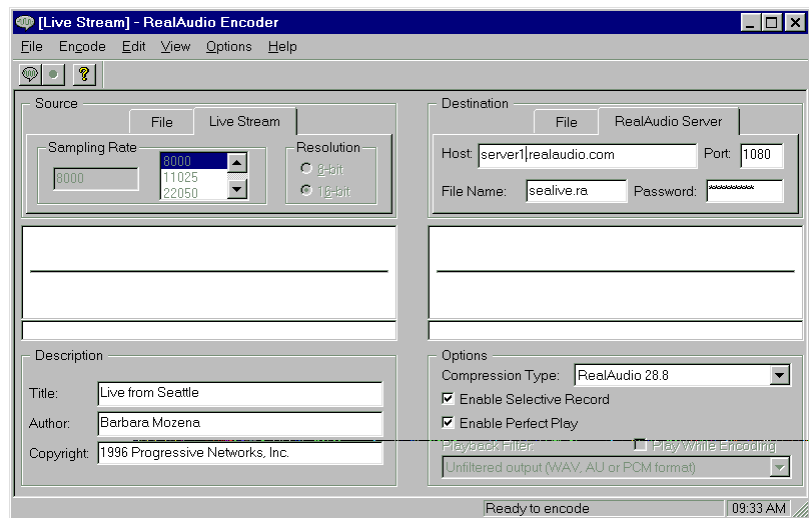
1. Install RealAudio encoder on the encoding machine.
2. Install RealServer on the serving machine. (It may be the same as the encoding machine.)
3. Set up your preferred audio device to your audio card (not the video card).

RealAudio Encoder and RealServer computers can be on different platforms.

Live RealAudio Encoder for Windows

RealAudio Encoder for Windows runs on Windows 95 and Windows NT.

To provide content in multiple compression algorithms, run multiple machines using the RealAudio Encoder with the same input signal. Both copies of RealAudio Encoder should use identical settings except for **Compression Type** and stream name.



To encode and broadcast live content in Windows:


1. Click the **Live Stream** tab in the Source frame.
2. In the Description pane, enter the Title, Author, and Copyright information for your RealAudio output. These fields are optional.
3. If you want to allow RealPlayer Plus users to save your RealAudio signal to disk, select **Enable Selective Record**.

Note Enable PerfectPlay is not available for live broadcasts.

4. In the Destination pane, click the **RealAudio Server** tab.
5. In the **Host** box, type the domain name or the IP address of the RealServer computer.

6. In the **Port** box, type the port number from the PnaPort configuration setting in the RealServer configuration file.
7. In the **File Name** box, type a name for the live clip being encoded. This is the filename part of the URL that goes in the metafile (.ram) used to access the live broadcast.
8. In the **Password** box, type the password from the Encoder Password configuration setting in the RealServer configuration file.
9. In the Compression pane, select the appropriate compression type.

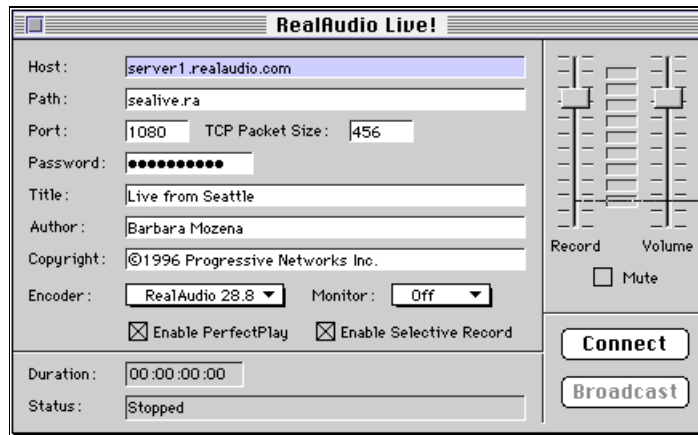
Note You can change Encoder default settings on the Preferences window.

10. Click  or the **Start Encoding** button to begin encoding the input and sending the data to the specified RealServer.

Live RealAudio Encoder for Macintosh

RealAudio Live Encoder for Macintosh runs on System 7.1 and later.

To provide multiple compression types, run multiple Encoder machines with the same input signal. All copies of RealAudio Encoder should use identical settings except for **Encoder**.



To encode and broadcast live content for Macintosh:

1. In the **Host** box, type the domain name or the IP address of the RealServer computer.
2. In the **Path** box, type a name for the live clip being encoded. This is the filename part of the URL that goes in the metafile (.ram) used to access the live broadcast.
3. In the **Port** box, type the port number from the PnaPort configuration setting in the RealServer configuration file.
4. In the **TCP Packet Size** box, keep the default value of 456.
5. In the **Password** box, type the password from the Encoder Password configuration setting in the RealServer configuration file.

6. Enter optional text for the **Title**, **Author**, and **Copyright** in the appropriate text boxes. This text is displayed by RealPlayer when a user listens to the broadcast.
7. Choose the **Encoder** value.
8. If you want to allow listeners with the RealPlayer Plus to save your RealAudio signal to disk, click **Enable Selective Record**.

Note Enable PerfectPlay is not available for live broadcasts.

9. Drag the **Volume** slider to adjust the monitor volume.
10. Drag the **Record** slider to set the encoding volume level for best quality.
11. Click **Connect** to connect the Encoder to RealServer.
12. Click **Broadcast** to begin encoding and sending data to RealServer.

Live RealAudio Encoding for UNIX

RealAudio Encoder for UNIX is run from the command line, using the following syntax:

```
raencoder options file
```

where `options` is the any of the RealAudio Encoder options described below and `file` is the input audio file to be compressed into the RealAudio format.

Note If no input filename is specified, STDIN is assumed.

RealAudio Encoder for UNIX displays a progress indicator on your screen as the file is encoded. If you plan to encode large numbers of files on a regular schedule, you can write a shell script to do batch encoding.

Option	Description																				
-a	Use this option to specify an Author string.																				
-c	Use this option to specify a Copyright string.																				
-f	<div>Use this option to specify a compression type. Valid options are:<table><tr><td>14_4 0</td><td>RealAudio 2.0 - 14.4</td></tr><tr><td>28_8 0</td><td>RealAudio 2.0 - 28.8</td></tr><tr><td>DNET 0</td><td>RealAudio 3.0 - 28.8 Mono, narrow response</td></tr><tr><td>DNET 1</td><td>RealAudio 3.0 - 28.8 Mono, medium response</td></tr><tr><td>DNET 2</td><td>RealAudio 3.0 - 28.8 Mono, full response</td></tr><tr><td>DNET 3</td><td>RealAudio 3.0 - 28.8 Stereo</td></tr><tr><td>DNET 4</td><td>RealAudio 3.0 - ISDN Mono</td></tr><tr><td>DNET 5</td><td>RealAudio 3.0 - ISDN Stereo</td></tr><tr><td>DNET 6</td><td>RealAudio 3.0 - Dual ISDN Mono</td></tr><tr><td>DNET 7</td><td>RealAudio 3.0 - Dual ISDN Stereo</td></tr></table><div>Default: 28_8 0</div></div>	14_4 0	RealAudio 2.0 - 14.4	28_8 0	RealAudio 2.0 - 28.8	DNET 0	RealAudio 3.0 - 28.8 Mono, narrow response	DNET 1	RealAudio 3.0 - 28.8 Mono, medium response	DNET 2	RealAudio 3.0 - 28.8 Mono, full response	DNET 3	RealAudio 3.0 - 28.8 Stereo	DNET 4	RealAudio 3.0 - ISDN Mono	DNET 5	RealAudio 3.0 - ISDN Stereo	DNET 6	RealAudio 3.0 - Dual ISDN Mono	DNET 7	RealAudio 3.0 - Dual ISDN Stereo
14_4 0	RealAudio 2.0 - 14.4																				
28_8 0	RealAudio 2.0 - 28.8																				
DNET 0	RealAudio 3.0 - 28.8 Mono, narrow response																				
DNET 1	RealAudio 3.0 - 28.8 Mono, medium response																				
DNET 2	RealAudio 3.0 - 28.8 Mono, full response																				
DNET 3	RealAudio 3.0 - 28.8 Stereo																				
DNET 4	RealAudio 3.0 - ISDN Mono																				
DNET 5	RealAudio 3.0 - ISDN Stereo																				
DNET 6	RealAudio 3.0 - Dual ISDN Mono																				
DNET 7	RealAudio 3.0 - Dual ISDN Stereo																				
-h	Use this option to display this list of options.																				

Option	Description
-H	Required for live broadcasting. Use to indicate the name of the host.
-i	<p>Use this option to display the information stored in the header of your input file. For example, typing:</p> <pre>raencoder -i file</pre> <p>might produce the output:</p> <pre>WAVE Header Encoding is linear; 2 bytes per sample Number of bytes of audio is 23038. Sampling rate: 8000 # of channels: 1</pre>
-l	<p>Required for live broadcasting. Use to indicate source of audio plugged into your sound card. Valid options are:</p> <pre>line cd mic</pre>
-L	Required for live broadcasting. Use to connect to RealServer.
-o	Use this option to specify the output RealAudio file name. If you do not use this option, then the output file is named file.ra where file is the name of the input file.
-p	<p>Use this option to enable or disable PerfectPlay. In PerfectPlay mode, the RealPlayer Plus uses can play higher bandwidth files over slow connections (for example 14.4 Kbps modem connections). Valid options are:</p> <pre>0 Disabled 1 Enabled</pre> <p>Default: 1</p>

Option	Description				
-P	Required for live broadcasting. Used to indicate the Port Number.				
-q	Use this option to disable the status indicator while encoding.				
-r	<p>Use this option to set the input-file sampling rate. This option overrides the sampling rate specified in the header of the input audio file.</p> <p>Valid values are 8, 11, 16, 22, 32, 44.</p> <p>Default: 8</p>				
-s	<p>Use this option to specify the sample width of .pcm input audio data. Valid options are:</p> <table><tr><td>8</td><td>8-bit .pcm data</td></tr><tr><td>16</td><td>16-bit .pcm data</td></tr></table> <p>Default: 8</p>	8	8-bit .pcm data	16	16-bit .pcm data
8	8-bit .pcm data				
16	16-bit .pcm data				
-S	<p>Required for live broadcasting of an external audio source. Used to indicate the output file size to reach before stopping encoding.</p> <p>Valid format: xxxxx KB</p>				
-t	Use this option to specify a Title string. The string is displayed in the RealAudio Player window.				
-T	<p>Required for live broadcasting of an external audio source. Used to indicate the time to stop encoding.</p> <p>Valid format: DD:HH:MM:SS</p>				
-v	Use to display the version number of RealAudio Encoder.				

Option	Description
-V	Optional for live broadcasting of an external audio source. Use to scale the input volume. Range: 1 (lowest) to 100 (highest).
-w	Use this option to enable or disable Selective Record. Enabling Selective Record allows RealPlayer Plus users to save your clip. Valid options are: 0 Disabled 1 Enabled Default: 0
-W	Required for live broadcasting. Use to specify the Password.

The following are examples of the command line for live broadcasting:

Encoding and broadcasting a live file with the RealAudio 3.0 - 28.8 Mono medium response algorithm:

```
raencoder -t"My Title" -a"My Name" -c"My  
Company, 1996" -omyfile.ra -l"line" -H"Hostname"  
-P"portnumber" -W>Password"
```

Archiving Live Broadcasts

You can choose to archive a live broadcast for playback later. RealServer can be configured to automatically archive live broadcasts or you can use the **rmfile** or **rafile** utility program to archive broadcasts on any RealServer over a network. You can choose to create just one file; a new file based on elapsed time, such as every 30 minutes; or a new file based on size, such as every 5 MB.

If you specify the **LiveFileTarget** and **LiveFilePassword** settings in the server's configuration file, RealServer automatically archives any live media stream that arrives at the Server. These media streams are archived as RealMedia (.rm) files. To archive an audio only stream as a RealAudio file, use **rafile**. Be sure you have enough available disk space to store the files generated from a live broadcast. The archive files are stored in the directory specified by the **LiveFileTarget** parameter, or in the working directory that was used to start RealServer if a target directory is not specified.

Because the **rmfile** or **rafile** utility program accepts a network address for the media source, you do not need to run it on the same computer as RealServer, RealAudio Encoder, or RealVideo Encoder. Archive files written by **rmfile** or **rafile** are stored in the specified directory, or in the working directory used to start the **rmfile** or **rafile** program if no directory is specified.

If RealServer or **rmfile** archives a live broadcast with the same destination path and file name used for a previous broadcast, the .rm file from the previous broadcast is overwritten. Reusing the same output file name can simplify Web page maintenance, because the links for a recurring event remain the same. If you want to maintain an archive of live broadcasts, either copy the .rm file elsewhere before it is overwritten or use unique file names for each live broadcast.

Example 1:

A television station broadcasts over the Internet and wants to archive the entire broadcast day in multiple files each 60 minutes long to the **/usr/Archive** directory. The broadcast is named Live.rm and is available in RealVidex 28.8 format only. The configuration parameters are:


```
LiveFilePassword rmBroadcast1
LiveFileTarget    /usr/Archive
LiveFileTime      1h
```

RealServer automatically archives the live broadcast to a series of files in the /usr/Archive directory named Live0.rm, Live1.rm, Live2.rm, and so on. Each file contains one hour of audio data encoded in the RealVideo 28.8 format.

Example 2:

A concert promoter broadcasts a live concert over the Internet and wants to archive the entire concert.

Instead of archiving on the Server computer, a separate archive computer is used. The archive computer runs the **rmfile** utility program from a command line.

The configuration parameters on the Server computer are:

```
LiveFilePassword rmBroadcastZ
```

The command on the archive computer is:

```
rmfile -b -p rmBroadcastZ
pnm://server.domain.com/Concert.rm Bconcert.rm
```

The **rafile** utility program connects to the Server using the URL pnm://server.domain.com/Concert.ra. These files can be copied to a RealServer for later rebroadcast of the concert.

For additional information on the **rmfile** command and its settings, see “rmfile” on page 169.

Rmfile can be driven by either command line options or by a configuration file. The command line options will always overrule the configuration file. The configuration entry details are

LiveFilePassword

Password used for archiving live broadcasts. Used by RealServer for automatic archiving and by the **rmfile** or **rafile** utility program.

Used by	pnserver, rmfile, rafile
Default value	(none)
Range of values	alphanumeric string without spaces
Restart Server after change	no

```
LiveFilePassword <password>
```

For more information, see “rmfile” on page 169 and “rafile” on page 171.

Example

```
LiveFilePassword rmBroadcast1
```

LiveFileSize

Size of file, in megabytes, used for creating archive files of live broadcasts.

Used by	pnserver, rmfile, rafile
Default value	0
Range of values	Integers greater than or equal to zero
Restart Server after change	No

By default, **rmfile** or **rafile** uses this setting unless overridden with the **-s** option.

```
LiveFileSize <value>
```

For more information, see “rmfile” on page 169 and “rafile” on page 171.

Example

```
LiveFileSize 5
```

LiveFileTarget

File or directory to use to create the archive files of live broadcasts.

Used by	pnserver, rmfile, rafile
Default value	(none)
Range of values	Valid file name
Restart Server after change	No

By default, **rmfile** or **rafile** uses this value unless overridden by a file or directory name on the command line.

```
LiveFileTarget <name>
```

If name is a directory name, **rmfile** or **rafile** uses the filename of the live broadcast to name files. If it is a filename, it creates files in the working directory used to start **rmfile** or **rafile** or RealServer. In either case, it appends numbers to the archive files, starting at 0.

For more information, see “rmfile” on page 169 and “rafile” on page 171.

Example

```
LiveFileTarget pnfm.rm
```

Makes **rmfile** create archive files named **pnfm1.rm**, **pnfm2.rm**, and so on.

```
LiveFileTarget /usr/evand/rmfiles
```

Makes **rmfile** create archive files in the directory **/usr/evand/rmfiles** and names files using the filename list in the URL setting.

LiveFileTime

Maximum length, in time, of a archive file of a live broadcast.

Used by	pnservice, rmfile, rfile
Default value	0
Range of values	Integers greater than or equal to zero and letters d, h, and m.
Restart Server after change	No

By default, **rmfile** or **rfile** uses this setting unless overridden with the **-t** option.

```
LiveFileTime <value>
```

Specify time as a number and letter, such as 1m for one minute, 1h for one hour, and 1d for one day.

For more information, see “rmfile” on page 169 and “rfile” on page 171.

Example

```
LiveFileTime 1h
```

URL

URL that points to the live media stream to be recorded by rfile.

Used by	rmfile, rfile
Default value	(none)
Range of values	valid URL
Restart Server after change	No

rmfile or **rafile** uses this setting unless overridden by a URL on the command line.

URL <url>

For more information, see “rmfile” on page 169 and “rafile” on page 171.

Example

URL pnm://server:7071/live1.rm

BandwidthEncoding

Specifies the default bandwidth for archive files of live broadcasts. Not required for rmfile.

Used by	rafile
Default value	None
Range of values	14_4.18, dnet.20, dnet.25, 28_8.36, dnet.50, dnet.100
Restart Server after change	No

By default, **rafile** uses this setting unless overridden with the **-e** option.

BandwidthEncoding <value>

For more information, see “rafile” on page 171.

Example

BandwidthEncoding dnet.20

Note **Rmfile** is the main utility and supports .rm files; .ra files were used in the RealAudio 3.0 Server.

LiveFileBandwidthNegotiation

Specifies that the rfile program use bandwidth-negotiation style of naming for archive files of live broadcasts. Not required for rmfile.

Used by	pnserver, rfile
Default value	False
Range of values	True, False
Restart Server after change	No

By default, **rfile** uses this setting unless overridden with the **-b** option.

```
LiveFileBandwidthNegotiation <value>
```

For more information, see “rfile” on page 171.

Example

```
LiveFileBandwidthNegotiation TRUE
```

Note **Rmfile** is the main utility and supports .rm files; .ra files were used in the RealAudio 3.0 Server.

Simulating a Live Broadcast

At times, you might want to play a recorded media file as if it were being broadcast live. Perhaps you want to test your system before a live event, delay broadcast of a concert, or play a commercial on your site. The **slta** (Simulated Live Transfer Agent) utility lets you play a recorded media file as if it were live.

To use **slta**, you need to specify the password from the Server's **EncoderPassword** configuration parameter and the name of the input and output file.

Simulated Live Broadcast

The **rvslta** (RealVideo simulated live transfer agent) utility delivers pre-recorded content as if it were a live event. Users connecting to the link will get the event “in progress”. It can be used as a test, to delay broadcast of a live event or to multicast pre-recorded content. The syntax is as follows:

```
rvslta -i path\infile.rm /o outfile.rm -s server [-p  
port] [-w password]
```

where:

- infile** is the path and file name to the input file.
- outfile** is the path and file name to the output file.
- server** is the server name.
- port** is the server port (port defaults to 7070).
- password** specifies the password slta uses to connect to the server (defaults to none).

For example:

```
rvslta -i c:\livefile.rm -o livenow.rm -s  
www.testserver.com
```

InputFile

The path of a file to convert to a live file.

Used by	rvslta
Default value	(none)
Range of values	Valid file name
Restart Server after change	No

By default, **slta** uses this value unless overridden by a filename on the command line.

```
InputFile <filename>
```

For more information, see “slta” on page 177.

Example

```
InputFile /usr/cnfn/show1.ra
```

OutputFile

Name of the simulated live stream sent using the **slta** utility.

Used by	slta
Default value	(none)
Range of values	Valid media file name
Restart Server after change	No

slta uses this setting unless overridden by a filename on the command line.

```
OutputFile <filename>
```

For more information, see “slta” on page 177.

Example

```
OutputFile broadcast.ra
```

ServerHost

Name of a RealServer to receive the live file.

Used by	slta
Default value	(none)
Range of values	Valid DNS name
Restart Server after change	N/A

The **slta** utility uses this setting unless overridden by a host name on the command line. For more information, see “slta” on page 177.

```
ServerHost <host>
```

Example

```
ServerHost server1.real.com
```

ServerPassword

Password that slta must use to connect to RealServer.

Used by	slta
Default value	(none)
Range of values	Alpha-numeric string without spaces
Restart Server after change	Yes

By default, **slta** uses this value unless overridden by a password on the command line. For more information, see “slta” on page 177.

```
ServerPassword <password>
```

Example

```
ServerPassword StreamKey1
```

ServerPort

Number of port on the RealServer to receive the live file from **slta**.

Used by	slta
Default value	(none)
Range of values	Valid port number
Restart Server after change	Yes

Must be the port number of the RealServer specified by the **ServerHost** setting. **slta** uses this setting unless overridden by a port number on the command line.

```
ServerPort <port>
```

For more information, see “slta” on page 177.

Example

```
ServerPort <port> 8081
```

Multicasting

This chapter provides a brief overview of multicasting and describes how to configure RealServer and RealPlayer for multicasting.

Overview

Multicasting is a way of sending a live broadcast as a single data stream to only those clients who request the data. Multicasting contrasts with two other modes of transmission:

- Unicasting, which sends a separate, point-to-point data stream to each client that requests it.
- Broadcasting, which sends a single data stream to all clients on a subnet, regardless of whether a client has requested the data. (Note that broadcasting in this sense is not the same as broadcasting a live event from RealServer. Broadcasting a live event means transmitting the event live—it does not indicate the mode of transmission.)

To reach a set of clients that request a particular data stream, multicasting consumes much less bandwidth than either unicasting or broadcasting. Multicasting sends only a single data stream (rather than many copies of the data stream) to only those clients who request the data (rather than all clients on a subnet).

Multicast delivery over the Internet is made possible to a limited degree by the Internet Multicast Backbone (Mbone), which is a virtual network consisting of those portions of the Internet that are multicast-enabled. For the most part, however, multicasting is used over intranets.

Multicast Transmission

A server transmitting a live media event using multicast sends UDP multicast datagrams to a single IP address. This multicast address represents a particular transmit session rather than a specific client's address. A multicast address is also called a host group address since, in effect, it also represents a group of clients (hosts) that have requested to receive the broadcast. Multicast addresses can range from 224.0.0.0 to 239.255.255.255.

Multicast datagrams are sent using the same IP operations as UDP unicast datagrams. In essence, a multicast broadcast is a unicast broadcast sent to a multicast address.

Each multicast packet contains a time-to-live (TTL) field in its IP header. The TTL field limits the scope of the broadcast by limiting the number of times that the packet can be forwarded by a router. The TTL setting is often used to limit the scope of a broadcast to a particular intranet.

Multicast Reception

Multicast reception is much more complicated than multicast transmission. The following description focuses on how RealPlayer and RealServer interact to establish a multicast connection:

1. When a user requests to receive RealAudio or RealVideo media, RealPlayer opens a TCP/IP connection to RealServer. RealPlayer indicates that it is multicast-enabled.
2. If the media is a live broadcast stream, RealServer sends back a message over the TCP/IP connection that notifies RealPlayer of the multicast address and the multicast port. If RealServer is not already sending multicast packets to the multicast address, it scans the address range specified in **MulticastAddressRange** (see “MulticastAddressRange” on page 320) for the first available address. RealServer sends the live broadcast to the first available multicast address that it finds within the range.
3. RealPlayer creates two UDP connections. On one connection, it sends an Internet Group Management Protocol (IGMP) query that indicates to its

nearest router that it wants to join the multicast host group at the multicast address and port specified by RealServer.

4. One or more routers, which must be multicast-enabled, then route the multicast packets to the subnet on which RealPlayer is located.
5. The computer on which RealPlayer is running starts filtering the network for packets sent to the multicast address and passes these packets to RealPlayer.

By default, RealPlayer is configured for various fall-back connection options if a multicast connection cannot be established with RealServer (see “Configuring RealPlayer for Multicasting” on page 318):

1. If, after sending out an IGMP query, RealPlayer does not receive any multicast packets after a specified number of seconds, it breaks the TCP/IP connection with RealServer
2. RealPlayer then re-establishes a new TCP/IP connection with RealServer. This time, however, RealPlayer indicates that it will not receive multicast broadcasts and expects to receive UDP unicast packets.
3. If RealPlayer does not receive a UDP unicast packet after a specified number of seconds, it again breaks the TCP/IP connection with RealServer and re-establishes a new TCP/IP connection with RealServer. This time, RealPlayer indicates that it wants to receive the media over the TCP/IP connection, which is a more secure (in terms of packet loss) but slower connection.

Multicast Requirements

To support multicasting, both the sending and receiving nodes, as well as the routers between the sending and receiving nodes, must be multicast-enabled. The details of these requirements are beyond the scope of this chapter. This chapter describes only what is required to enable RealServer and RealPlayer for multicast broadcasting and multicast reception, respectively.

Multicast delivery of RealAudio or RealVideo requires:

- Multicast-enabled client computers running RealAudio Player 3.0 or RealPlayer. The TCP/IP protocol stack on the client computer must

support multicast reception. The network adapter and driver on the client computer must be able to filter for data link layer addresses mapped from network-layer multicast addresses.

- A correctly configured RealServer running on a computer that is correctly configured for multicast support. The TCP/IP protocol stack on the computer must support multicast transmission.
- Any routers on the client network must be multicast enabled.

To receive audio-only broadcasts using multicast, the client must use RealAudio Player 3.0 or RealPlayer. To receive video broadcasts using multicast, the client must use RealPlayer.

Combining Splitting and Multicasting

To reach large audiences across the Internet, use splitters (see page 318) to send data across the Internet, and then use multicast delivery within each target intranet.

Configuring RealPlayer for Multicast Reception

For a client to receive a multicast broadcast, it must request and be granted membership in the host group to which the multicast broadcast is being sent. By default, RealAudio Player 3.0 and RealPlayer are enabled to request multicast transmission of live broadcasts.

To check whether a RealAudio Player 3.0 for Windows or a RealPlayer for Windows is enabled for multicasts:

1. From the View menu, choose Preferences.
2. Click the **Transports** button.
3. Click **Use specified transports**.
4. Click the **Specify transports** button.
5. Verify that **Attempt to use Multicast for live content** is selected.

Configuring RealServer for Multicast Broadcasting

To enable multicast delivery:

1. Verify with your network administrator that the routers in your network are multicast-enabled and that the computer running RealServer is correctly configured for multicast support.
2. Specify the range of multicast destination addresses (host group addresses) available to RealServer, using the **MulticastAddressRange** configuration setting. The network administrator should know which multicast addresses are available on the intranet. Be sure to include enough addresses in the range to accommodate your multicast broadcasting needs. See “MulticastAddressRange” on page 320 for details.
3. Specify the addresses of client computers or networks that are allowed to connect to multicast broadcasts from RealServer. Use the **MulticastControlList** configuration setting to do this. Clients whose addresses are not included in the **MulticastControlList** can connect to RealServer only in unicast mode. See “MulticastControlList” on page 321 for details.
4. Specify how far multicast packets can travel in your network. Use the **MulticastTTL** configuration setting to do this. The default value of 16 keeps multicast packets within a typical internal network at a site. See “MulticastDeliveryOnly” on page 323 for details.

To Specify Optional Multicast Settings

1. If you want to limit clients and networks specified in **MulticastControlList** to multicast connections only, set the **MulticastDeliveryOnly** configuration setting to True. This is an optional setting that can help reduce bandwidth usage on an intranet. A client whose address is not in the **MulticastControlList** can still make unicast connections to RealServer. See “MulticastDeliveryOnly” on page 323 for details.
2. If you want to specify a TCP/IP port for multicasting that is different than the default port (7070), set the **MulticastPort** configuration setting. You

should not need to change this setting unless you are having problems. See “MulticastPort” on page 323 for details.

MulticastAddressRange

The range of multicast addresses (host group addresses) to which RealServer can send a live multicast stream. The network administrator should know which multicast addresses are available on the local intranet.

When starting a multicast delivery of a live broadcast, RealServer scans the address range specified in **MulticastAddressRange** for the first available address. RealServer sends the live broadcast to the first available multicast address that it finds within the range.

Be sure to include enough addresses in **MulticastAddressRange** to accommodate your broadcasting needs. If you are running only one RealServer process (see **MaxThreads** on page 84), you need one multicast address for each live multicast broadcast. For example, if you are broadcasting three live programs by multicast, you need at least three addresses in the **MulticastAddressRange**. If you are running more than one RealServer process, you should make available one multicast address for each broadcast times the number of forks used by RealServer. For example, if you are broadcasting three live programs via multicast, and RealServer is using three forks, you need at least nine multicast addresses.

Multicast addresses can range from 224.0.0.0 to 239.244.255.255. The addresses between 224.0.0.0 and 224.0.0.255 are reserved for routing protocols and other protocols. Other addresses and ranges are reserved for other applications. See RFC 1700, “Assigned Numbers,” for a complete list of restricted addresses. In general, if the multicast address does not start with the decimal numbers 244, it should be usable. **MulticastAddressRange** is required for multicast support.

Used by	pnservice
Default value	(none)
Range of values	Valid IP addresses in the range 224.0.0.0 - 239.255.255.255
Restart Server after change	No

```
MulticastAddressRange <address>-<address>
```

where:

<address> is an IP address configured for multicast delivery.

Example

```
MulticastAddressRange 230.125.141.0-230.125.141.255
```

MulticastControlList

The range of client or network addresses (typically on an intranet) which RealServer allows to receive multicast delivery. RealServer gives a client whose address is within the range access to multicast delivery only when the client requests such delivery.

This **MulticastControlList** is required for multicast support.

Used by	pnserv
Default value	(none)
Range of values	Valid IP addresses
Restart Server after change	No

```
MulticastControlList [{<address>, <net mask>}, ...]
```

where:

<address> is the domain address of the client computer or network for which RealServer uses multicast delivery if requested by the player.

<net mask> is a mask that specifies the bits in the domain address that are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards. For example, an address of 121.23.101.0 with a net mask of 255.255.255.0 accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted.

If you want to grant multicast access to all addresses within a domain, you can specify a net mask of 0.0.0.0 if you have a RealServer licensed for the Internet. If you have an intranet-only RealServer, you cannot specify a net mask of 0.0.0.0.

To prevent all Players from accessing multicast delivery, do not include a **MulticastControlList** value in your configuration file.

The net mask 255.255.255.255 accepts only the single IP address specified in the address.

Example

```
MulticastControlList [{204.71.154.0, 255.255.255.0}]
```

MulticastDeliveryOnly

When set to True, restricts the client addresses specified in **MulticastControlList** to multicast reception only—client addresses so restricted cannot connect to RealServer in unicast mode. This option can be used to help control bandwidth on a network (typically on an intranet).

MulticastDeliveryOnly does not affect client addresses that are not in the **MulticastControlList**—such addresses can connect to RealServer in unicast mode.

This **MulticastDeliveryOnly** setting is optional for multicast support.

Used by	pnserver
Default value	False
Range of values	True, False
Restart Server after change	No

MulticastDeliveryOnly <value>

Where:

<value> is True to restrict client addresses in the **MulticastControlList** to multicast reception only, or False to allow any type of reception.

Example

MulticastDeliveryOnly True

MulticastPort

The port number for multicast broadcasts. The **MulticastPort** setting is optional for multicast support.

Used by	pnservice
Default value	7070
Range of values	Any valid port number
Restart Server after change	No

```
MulticastPort <value>
```

Where:

<value> is the port number for multicast broadcasts.

Note that the multicast port and PNAport (see page 85) are used differently from each other. The multicast port is the port to which RealServer sends live multicast broadcasts. The PNA port is the port that a Player uses to establish a TCP/IP control connection with RealServer.

Example

```
MulticastPort 7075
```

MulticastTTL

The Time To Live (TTL) for multicast packets. This value is used by routers in your network to determine whether a multicast packet is allowed to pass through the router.

This **MulticastTTL** setting is optional for multicast support.

The following are the typical TTL values and their meanings:

TTL Value	Keep Packets Within
0 or 1	Local host
16	Site
63	Region
129-255	World

For most multicast uses, you should keep the multicast packets within your intranet by setting **MulticastTTL** to 16 or less.

See your network administrator for information on how your network is configured.

Used by	pnservice
Default value	16
Range of values	0 - 255
Restart Server after change	No

MulticastTTL <value>

where:

<value> is the TTL value included in multicast packet headers.

Example

```
MulticastTTL 16
```

Configuring Your Web Site

After encoding your RealAudio and RealVideo files, you are ready to attach the files to Web pages. The following sections explain the construction and use of RealAudio and RealVideo content on your Web site. When you have your RealServer set up, use this information to showcase audio and video content from your site.

Making the Most of Your Content

To get the most out of your RealAudio and RealVideo content, educate your Web site's visitors about RealAudio and RealVideo. Let people know that they can listen and view your clips instantly, without download delays. Identify each clip with a Real bubble icon to distinguish it as real-time audio and video.



Real bubble icon

You may obtain the above graphic from *How to Design a RealMedia Site* at:

<http://www.real.com>

Capture the graphic from within your Web browser by right-clicking (Windows) or Control-clicking (Macintosh) it and saving it to file.

Make it easy for your visitors to get RealPlayer by providing a link to the Progressive Networks home page at:

<http://www.real.com/>

Metafiles

HTML documents use hyperlinks to connect Web pages. RealAudio and RealVideo material is also reached via links. However, RealAudio and RealVideo links you put into your HTML pages are not direct references to the RealAudio or RealVideo files. Instead they are references to metafiles which contain information needed to establish a connection between your RealServer and your listener's RealPlayer and to initiate playback.

Metafiles contain the addresses of RealAudio (.ra) and RealVideo (.rm) files. These addresses are in the form of URLs. They begin with a locator type, followed by a specific address. The locator type identifies the protocol used to exchange information between client and server. Common locator types include http and ftp.

The locator type used by Progressive Networks software is pnm (Progressive Networks Metafile). Addresses you put into a metafile begin with `pnm: / /`

To create a link between two of your Web pages, you might add the following text to an HTML document:

```
<A HREF="sect2.htm">Continue to Section 2</A>
```

A visitor to your site could click the text, and your Web server would then deliver the appropriate section of your document.

If you employed the same syntax to create a link to a RealAudio and RealVideo file, your visitor's click would prompt your Web server to deliver your clip. But, your Web server cannot stream RealAudio and RealVideo for real-time playback. Only your RealServer can do that.

Your Web server can, however, convey information to a user's RealPlayer that enables that Player to establish a direct connection to your RealServer. Once in place, this direct connection is used to stream to the Player and to carry commands (such as seek or pause) back to your RealServer.

Instead of pointing to a file, your Web page is set up to point to a metafile. This metafile, in turn, contains the URL (Uniform Resource Locator) of the file (or files) you want associated with the hyperlink. The user's browser passes the URLs to RealPlayer, which retrieves .ra or .rm files from your RealServer.

Creating Metafiles

To create a metafile:

1. Use a text editor (such as Notepad) to create a file containing the RealAudio or RealVideo URL. The contents of your file should be in the following form:

```
pnm: //hostname/path
```

For example, to provide access to a RealVideo file called `hello.rm`, the text of your metafile would be:

```
pnm: //www.server1.com/hello.rm
```

where `www.server1.com` is the DNS name of the machine running your RealServer, on which you store your RealAudio or RealVideo files.

If you want more than one file to play when the user clicks your link, create a metafile containing several URLs (on separate lines with no intervening blank lines). For example, if your metafile contains:

```
pnm: //www.server1.com/hello.rm  
pnm: //www.server1.com/welcome.rm  
pnm: //www.server1.com/coolstuff.rm
```

the Player automatically plays your three files in sequence. A listener can use the Clip menu on the Player to move forward and backward between clips (.rm files).

2. Save your metafile in the “All files (*.*)” format, using a .ram file name extension.

For example, you could save the three lines shown above in a file named `3track.ram` on your Web server.

3. In your HTML document, reference the metafile in a hyperlink, followed by a reference to the Real icon (so that the Real icon is displayed to the left of the file) as follows:

```
<A HREF="http://www.real.com/welcome.ram"> <IMG  
SRC="http://www.real.com/pics/rvfile.gif"  
align=left border=0> Welcome!</A>
```

where `welcome.ram` is the metafile.

File Name Extensions

Each metafile that you create must be saved with (or renamed to have) a file name extension. This extension tells your Web server what the metafile is, to ensure that the enclosed URL is handled properly.

There are two metafile types: .ram and .rpm. These different file name extensions are passed on by your Web server and, ultimately, tell the user's Web browser which application to launch to play the referenced file:

.ram file – Browser launches RealPlayer

.rpm file – Browser launches RealPlayer Plug-in (see below)

Note You must configure your Web server to understand that the extension .ram refers to the MIME type **x-pn-realaudio** (detailed instructions for configuring a variety of Web servers are available TBD).

Customizing Calls to Video and Audio Content

Optional arguments may be added to metafiles to finesse what is seen and heard by users when they click your RealAudio or RealVideo link. You may alter the point in a clip at which play starts or ends, or the Title, Author, and Copyright information is displayed by RealPlayer.

Add the options to your metafile following the URL to which they apply. Options must be preceded by a ? (question mark) and separated from each other by an & (ampersand). The syntax is as follows:

```
pnm://www.real.com/test.rm?[opt1]&[opt2]
```

Changing Start and Stop Times

To create a link that starts playing a clip from a point other than the beginning of the file, use the **start** command. Specify the time into the clip at which play should begin. For example:

```
pnm://www.real.com/test.rm?start="30"
```

would result in playback starting thirty seconds into the file.

The format for the start time is as follows:

```
start="dd:hh:mm:ss.ss"
```

Tenths of seconds are separated from seconds by a decimal point; the other units of time are separated by colons. The time is interpreted from right to left,

and it is not necessary to specify days, hours, or minutes if these are not relevant.

Similar to the **start** option is the **end** option. For example, the metafile text:

```
pnm://www.real.com/test.rm?end="5:30"
```

is used to provide for playback of test.rm that terminates five minutes and thirty seconds from the start of the clip.

Note The end time is always measured from the actual start of the data in the file, even in the case where playback begins elsewhere. For example, the line:

```
pnm://www.real.com/test.rm?start="30"&end="5:30"
```

is used to start play of test.rm from the thirty-second mark and to stop play five minutes later.

Changing Title, Author, or Copyright Information

The following options exist to change the descriptive information from within the metafile:

```
title="new title"  
author="new author"  
copyright="new copyright"
```

Strings can be changed independently or in combination.

Changing information in this manner does not change what is stored in your file—only what is displayed when it is accessed through this particular metafile. This functionality is especially useful if, for example, you have one large file that contains your band's entire CD, and you want to credit the author of each song as it plays. You could create a multi-clip .ram file as follows:

```
pnm://www.server/band.rm?end="5:30"&title="Song1"  
pnm://www.server/band.rm?start="5:31"&end="7:45"&  
title="song2"&author="Joe Smith"  
pnm://www.server/band.rm?start="7:46"&end="15:01"  
&title="song3"&author="Jane  
Smith"&copyright="My Music, 1996"
```

HTTP Streaming

HTTP streaming enables content providers to stream RealAudio and RealVideo clips from a World Wide Web server. While this method is not as robust, it provides a reasonable method for providing short RealAudio and RealVideo content to a limited number of users.

Before you can stream RealAudio and RealVideo clips through HTTP, you must define the following MIME types for your World Wide Web server:

audio/x-pn-RealAudio (files with a .ra, .rm or .ram file extension)
audio/x-pn-RealAudio-plugin (files with a .rpm file extension)

Some World Wide Web servers are pre-configured with these MIME types.

Note If you are running a Web page off an ISP server, send mail to the ISP administrator asking them to configure RealPlayer mime type.

To stream RealAudio and RealVideo content using HTTP:

1. Copy your encoded RealAudio and RealVideo files (files with the .ra or .rm extension) to your World Wide Web server.
2. Use a text editor (such as Notepad) to create a metafile containing the RealAudio or RealVideo URL. For example, the contents of your file should be in the following form:

```
http://hostname/path
```

where hostname is the name of your World Wide Web server. For example: www.real.com

Note Refer to “Creating Metafiles” on page 328. This file works similarly, except it uses http as the protocol instead of pnm.

3. Save your metafile as “All Files (*.*)” using a .ram file name extension.
4. In your HTML document, reference the metafile in a hyperlink. For example:

```
<A HREF="file.ram">  
<A HREF="http://hostname/file.ram">
```

You can use relative or complete paths. If you use complete paths, you must include both the hostname and the complete path. For example:

```
<A HREF="http://www.real.com/home/welcome.ram">
```

5. When a user clicks on the link, the audio or video file(s) begin to download. RealPlayer begins playing after a few seconds; it does not need to wait for the entire file to be downloaded.

Custom Controls for RealAudio and RealVideo

RealAudio and RealVideo enables seamless integration of RealPlayer Controls into your Web page layout. You can place individual interactive components, such as a play button or image window, anywhere on your page, just as you would place an image using the `` tag in HTML.

There are two products which, in conjunction with the most popular Web browsers, enable “in page” audio controls.

- RealPlayer Plug-in provides Player-like features to browsers that support the Netscape Navigator Plug-in architecture. This plug-in also works in Internet Explorer 3.0.
- RealPlayer Control for ActiveX works with Internet Explorer 3.0 and Visual Basic applications to provide RealAudio and RealVideo playback capabilities.

Using RealPlayer Plug-in

The Plug-in runs as an adjunct to Web browsers that support Netscape’s Plug-in architecture. This RealPlayer Plug-in is included in the Player installation.

The `<EMBED>` tag specifies Plug-in attributes in HTML pages in much the same way that the `` tag specifies image attributes. The basic `<EMBED>` tag for RealPlayer contains only the attributes SRC, WIDTH, and HEIGHT, as shown below:

```
<EMBED SRC=metafile.rpm WIDTH=width_value  
HEIGHT=height_value>
```

For example:

```
<EMBED SRC="sample1.rpm" WIDTH=300 HEIGHT=134>
```

creates an in-page Player that is 300-pixels wide and 134-pixels high.

For the Plug-in, metafiles are stored with a .rpm file name extension.

The name tag is supported as an option with JavaScript to refer to a specific plug-in.

Note Do not place the <EMBED> tag within a table.

Feature	Description
SRC Attribute	<p>The SRC attribute specifies a metafile to be accessed. RealPlayer Plug-in is associated with a .rpm file name extension. This extension tells the user's Web browser to load RealPlayer Plug-in rather than the stand-alone RealPlayer.</p> <p>For the user's Web browser to correctly identify .rpm files, you or your system administrator must first configure the .rpm MIME type in your Web server. Users do not need to configure their Web browsers to recognize the .rpm MIME type. The plug-in architecture automatically sends .rpm files to RealPlayer Plug-in. Files with a .rpm extension are identical to .ram files, except for the extension.</p>

Feature	Description
WIDTH and HEIGHT Attributes	<p>The WIDTH and HEIGHT attributes specify the size of the embedded RealVideo component. Unlike images, Plug-ins do not size automatically. The WIDTH and HEIGHT can be specified in pixels (the default) or as a percentage of the Web browser window (for example: WIDTH=100%).</p> <p>Note If the WIDTH and HEIGHT attributes are not included, the Plug-in may appear as a tiny (and useless) icon with some browsers.</p> <p>If you want your Plug-in component to maintain an absolute size, specify HEIGHT and WIDTH in pixels. If you want the Plug-in graphic to scale with the Web browser window, specify size as a percentage. For example, if you want to fit the entire width of the Web browser window, use WIDTH=100%.</p>
CONTROLS Attribute	<p>The CONTROLS attribute of the <EMBED> tag allows you to place individual control elements within your page. You can use multiple <EMBED> statements to construct a custom interface, made up of individual controls. You can also place multiple controls within a single <EMBED> statement. CONTROLS supports the following values: All, ControlPanel, InfoVolumePanel, InfoPanel, StatusPanel, StatusBar, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, StatusField, ImageWindow.</p>
CONSOLE	<p>Sets a console name used to link multiple control instances. All controls with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same RealAudio or RealVideo clip. Call this function once for each instance of the Play or Stop button you want to link.</p> <p>The console name. “_master” links to all instances. “_unique” links to no other instances.</p>

Feature	Description
AUTOSTART	Sets whether or not the control automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
NOLABELS	Suppresses the Title, Author, and Copyright label text in the controls window. The text strings in the fields are still displayed.
RESET	Resets RealPlayer Control for ActiveX playlist. Valid values are TRUE or FALSE.
AUTOGOTOURL	Specifies how a URL is handled. Valid values are TRUE or FALSE. TRUE indicates that RealPlayer Control for ActiveX automatically forwards the URL event to the browser. FALSE indicates that the OnGotoURL VBScript event is used instead.

Preparing HTML Pages for Browsers that Cannot Use the Plug-in

Some Web browsers do not support plug-ins. You can create HTML pages that are enhanced for plug-ins but which also work for other browsers. Simply use the `<NOEMBED>` tag to include HTML statements for use by Web browsers that do not support Plug-ins.

The `<NOEMBED>` command should appear after an `<EMBED>` command and take the following syntax:

```
<NOEMBED> HTML to be ignored </NOEMBED>
```

For example, the command:

```
<EMBED SRC="sample1.rpm" WIDTH=300 HEIGHT=134>  
<NOEMBED> <A HREF="sample1.ram"> Play the clip  
using the stand-alone Player! </A></NOEMBED>
```

would show a page with the Plug-in if your page were accessed by a browser supporting Plug-ins, and would otherwise display the message “Play the clip using the stand-alone Player!” (and allow playback with the standard Player).

Using RealPlayer Control for ActiveX

You can embed RealPlayer Control for ActiveX in HTML pages using the Object tag.

The following is an example of RealPlayer Control <OBJECT> in an HTML page.

```
<OBJECT
  ID=RAOCX
  CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA"
  HEIGHT=140
  WIDTH=312>
  <PARAM NAME="SRC" VALUE="pnm://audio.real.com/file.rm">
  <PARAM NAME="CONTROLS" VALUE="all">
</OBJECT>
```

Note Directory names cannot have spaces.

Embedded Object Parameters

Feature	Description
OBJECT	Tag is used to embed RealPlayer Control for ActiveX. There should be one <Object> tag per RealPlayer Control on the page.
CLASSID	Specifies the control's CLSID. This value is always "clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA" for RealPlayer Control for ActiveX.
HEIGHT	Specifies the control's height on the HTML page. A value of 0 makes the control invisible.
WIDTH	Specifies the control's width on the HTML page. A value of 0 makes the control invisible.
PARAM	Specially embedded tag for supplying parameters to the ActiveX object.

The following properties are available with the PARAM Parameter:

Parameter	Description
SRC	Sets the source of RealAudio or RealVideo clip. The SRC location can be pnm, file or http protocol. This parameter is required.
CONTROLS	Sets the visible components of the control. Valid CONTROLS include All, ControlPanel, InfoVolumePanel, InfoPanel, StatusPanel, StatusBar, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, StatusField, and ImageWindow.
CONSOLE	<p>Sets a console name used to link multiple control instances. All controls with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same RealAudio or RealVideo clip. Call this function once for each instance of the Play or Stop button you want to link.</p> <p>The console name. “_master” links to all instances. “_unique” links to no other instances.</p>
AUTOSTART	Sets whether or not the control automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
NOLABELS	Suppresses the Title, Author, and Copyright label text in the controls window. The text strings in the fields are still displayed.
RESET	Resets RealPlayer Control for ActiveX playlist. Valid values are TRUE or FALSE.
AUTOGOTOURL	Specifies how a URL is handled. Valid values are TRUE or FALSE. TRUE indicates that RealPlayer Control for ActiveX automatically forwards the URL event to the browser. FALSE indicates that the OnGotoURL VBScript event is used instead.

Specifying How the Control Should Look

The **CONTROLS** attribute allows you to place individual control elements within your page. The **CONTROLS** attributes for the Netscape Navigator Plug-in and RealPlayer ActiveX Control are the same. The following explains the output of each attribute:

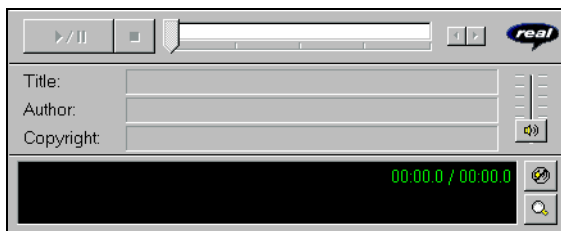
All - Displays a full Player view including the Control Panel, Information-and-Volume Panel and Status Bar.

Minimum Width: 21%

Maximum Width: 100%

Minimum Height: 23%

Maximum Height: 80%



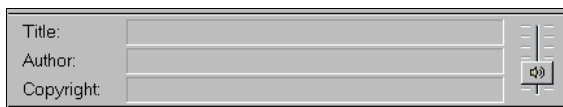
InfoVolumePanel - Displays the Title, Author, and Copyright information panel and the volume slider.

Minimum Width: 21%

Maximum Width: 100%

Minimum Height: 12%

Maximum Height: 50%



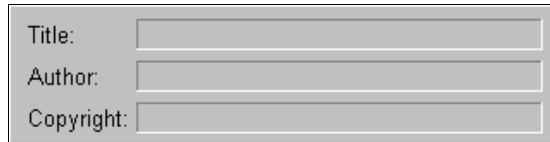
InfoPanel - Displays the Title, Author, and Copyright information.

Minimum Width: 20%

Maximum Width: 100%

Minimum Height: 10%

Maximum Height: 50%



A rectangular form with a light gray border. It contains three text input fields stacked vertically. The first field is labeled 'Title:', the second 'Author:', and the third 'Copyright:'. Each label is positioned to the left of its corresponding input field.

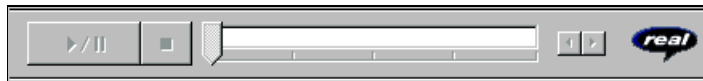
ControlPanel - Displays the play/pause button, the stop button and the position slider.

Minimum Width: 21%

Maximum Width: 100%

Minimum Height: 8%

Maximum Height: 25%



StatusPanel - Displays the Status Panel showing informational messages, current time position, and clip length. If you do not embed a Status Panel in your page, error messages are displayed in the Web browser's status bar.



PlayButton - Displays the play/pause button.

Minimum Width: 5%

Maximum Width: 100%

Minimum Height: 2%

Maximum Height: 25%



StopButton - Displays the stop button.

Minimum Width: 5%
Minimum Height: 2%

Maximum Width: 100%
Maximum Height: 25%



VolumeSlider - Displays the volume slider.

Minimum Width: 4%
Minimum Height: 10%

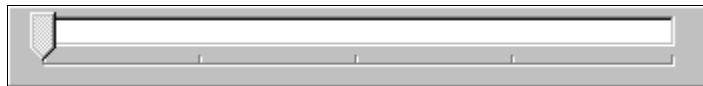
Maximum Width: 100%
Maximum Height: 100%



PositionSlider - Displays the position slider.

Minimum Width: 8%
Minimum Height: 5%

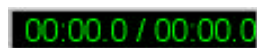
Maximum Width: 100%
Maximum Height: 25%



PositionField - Displays the field of the Status Bar showing position and clip length.

Minimum Width: 10%
Minimum Height: 4%

Maximum Width: 100%
Maximum Height: 25%



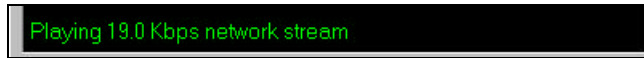
StatusField - Displays the message text area of the Status Bar.

Minimum Width: 13%

Maximum Width: 100%

Minimum Height: 4%

Maximum Height: 15%



ImageWindow - Displays the video image. (Only available for .rm files)

Minimum Width: 100%

Maximum Width: 100%

Minimum Height: 100%

Maximum Height: 100%



StatusBar - Displays the status field, position field, channels (stereo/mono).

Minimum Width: 21%

Maximum Width: 100%

Minimum Height: 5%

Maximum Height: 14%



ActiveX Methods and Properties

Methods

Methods are functions that control the performance of the Control. Unless otherwise noted, these methods have no return values and no parameters.

Method	Description
DoPlayPause	Plays or pauses the current clip. Equivalent to clicking the Play/Pause button.
DoStop	Stops the clip. Equivalent to clicking the Stop button.
DoNextItem	Skips to the next clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealAudio or RealVideo files.
DoPrevItem	Skips to the previous clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealAudio or RealVideo files.
CanPlayPause	Tests if Play/Pause function is available. Returns TRUE or FALSE
CanStop	Tests if Stop function is available. Returns TRUE or FALSE.
HasNextItem	Tests if the next clip function is available. The next clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the last clip in the .ram file. Returns true or false.
HasPrevItem	Tests if the previous clip function is available. The previous clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the first clip in the .ram file. Returns true or false.
AboutBox	Opens the Copyright information for the control.
EditPreferences	Opens the Preferences dialog box. Enables the end user to set his/her preferences. These preferences are global to all RealAudio or RealVideo clients on the machine.
HideShowStatistics	Shows or hides the Connection Statistics dialog box.
IsStatisticsVisible	Tests if the Connection Statistics dialog box is displayed. Returns true or false.
DoGotoURL(url, target)	Causes the control to attempt a navigation to the specified URL in the specified frame target. The container must support URL browsing. Parameters: string URL, string target.

Object Properties

The object properties are set within Visual Basic and specify properties about the control.

Property	Description
Source	Specifies the URL of the clip to play. The Source location can be pnm:, file: or http: protocol.
Controls	Returns/Sets the visible components of the control. Valid CONTROLS include ALL, ControlPanel, InfoVolumePanel, InfoPanel, StatusBar, StatusPanel, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, ImageWindow and StatusField.
Console	Sets a console name used to link multiple control instances. All controls with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same clip. Call this function once for each instance of the Play or Stop button you want to link. The console name master links to all instances while unique links to no other instances.
Autostart	Sets whether or not the control automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
NoLabels	Suppresses the Title, Author, and Copyright label text in the controls window. The text strings in the fields are still displayed.
AutoGotoURL	Specifies how a URL will be handled. Valid values are TRUE or FALSE. TRUE indicates that the RealPlayer ActiveX Control will automatically forward the URL event to the browser. FALSE indicates that the OnGotoURL VBScript event will be sent instead.

Java and JavaScript Methods

Methods are functions that control the performance of the Java enabled RealPlayer Plug-in.

Method	Description
SetSource(String <i>Source</i>)	Specifies the URL of the RealAudio clip to play. The Source location can be pnm:, file: or http: protocol.
SetControlsString(String <i>ControlsString</i>)	Returns/Sets the visible components of the control. Valid CONTROLS include ALL, ControlPanel, InfoVolumePanel, InfoPanel, StatusBar, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, and StatusField.
SetConsoleName(String <i>ConsoleName</i>)	Sets a console name used to link multiple RealVideo or RealAudio Plug-in instances. All Plug-ins with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same clip. Call this function once for each instance of the Play or Stop button you want to link. The console name master links to all instances while unique links to no other instances.
SetAutoStart(Boolean <i>bAutoStart</i>)	Sets whether or not the Plug-in automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
SetNoLabels(Boolean <i>bNoLabels</i>)	Suppresses the Title, Author, and Copyright label text in the Plug-in window. The text strings in the fields are still displayed.
DoPlayPause()	Plays or pauses the current clip. Equivalent to clicking the Play/Pause button.
DoStop()	Stops the RealAudio clip. Equivalent to clicking the Stop button.
DoNextItem()	Skips to the next clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealAudio files.
DoPrevItem()	Skips to the previous clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealVideo or RealAudio files.
CanPlayPause()	Tests if Play/Pause function is available. Returns TRUE or FALSE
CanStop()	Tests if Stop function is available. Returns TRUE or FALSE.
HasNextItem()	Tests if the next clip function is available. The next clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the last clip in the .ram file. Returns true or false.
HasPrevItem()	Tests if the previous clip function is available. The previous clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the first clip in the .ram file. Returns true or false.
AboutBox()	Opens the Copyright information for the control.
EditPreferences()	Opens the Preferences dialog box. Enables the end user to set his/her preferences. These preferences are global to all clients on the machine.
HideShowStatistics()	Shows or hides the Connection Statistics dialog box.
IsStatisticsVisible()	Tests if the Connection Statistics dialog box is displayed. Returns true or false.

CallBack Methods

CallBack methods are functions that provide notification from the RealAudio or RealVideo engine. CallBack methods can only be used with Java applets by implementing the `RAObserver` interface.

Method	Description
<code>onClipOpened(String <i>shortClipname</i>, String URL)</code>	Sent when a clip has been opened.
<code>onClipClosed()</code>	Sent to indicate that no clip is currently open.
<code>onShowStatus(String <i>status</i>)</code>	Sent to indicate that the status text is changing.
<code>setAutoGoToURL(boolean <i>bAutoGoToURL</i>)</code>	Specifies how a URL will be handled. Valid values are true or false. True indicates that the RealPlayer Plug-in will automatically forward the URL event to the browser. False indicates that the <code>OnGoToURL</code> event will be handled by the Java applet instead.
<code>onGoToURL(String <i>url</i>, String <i>target</i>)</code>	Sent when a URL event has been encountered for the currently playing clip. This event will only occur if the <code>AutoGotoURL</code> property is false.

Advanced Control Attributes

The more exciting features of RealPlayer Plug-in and RealPlayer Control for ActiveX may be utilized by specifying custom attributes within your HTML documents.

Removing Text Labels from Controls

If your Web page is in a language other than English, or if you want to use the Description fields to display information other than Title, Author, and Copyright, you may remove the Title, Author, and Copyright Labels from the information area. Controls that display Title, Author, and Copyright information for a clip, support a `NOLABELS=TRUE` attribute.

For example, allowing the default behavior

```
<EMBED SRC="use_lbl.rpm" WIDTH=350 HEIGHT=80  
CONTROLS=InfoPanel>
```

or specifying `NOLABELS=FALSE`

```
<EMBED SRC="use_lbl.rpm" WIDTH=350 HEIGHT=80  
CONTROLS=InfoPanel NOLABELS=FALSE>
```

results in the following display.

Title:	Welcome to RealVideo
Author:	The RealVideo Team
Copyright:	1997 Progressive Networks

On the other hand, using `NOLABELS=TRUE`, for example:

```
<EMBED SRC="no_lbl.rpm" WIDTH=350 HEIGHT=80  
CONTROLS=InfoPanel NOLABELS=TRUE>
```

produces the following InfoPanel:

Welcome to RealVideo
The RealVideo Team
1997 Progressive Networks

Starting Clips Automatically

Adding an AUTOSTART=TRUE attribute tells the user's browser to automatically begin playing your clip when the page is visited. You can use this feature to begin narration, to play a welcome message or start a video.

Since only one clip can play at a time, if you specify AUTOSTART for more than one control instance, only the last control to load will play. The order in which your files are delivered is dependent on the Web server and on the browser's cache size. This is not necessarily the order in which you put them within your HTML. Therefore, you should specify AUTOSTART for only one control instance per page.

Playing Clips with a Hidden Control or Plug-in

If you want to play RealAudio or RealVideo clips without having a visible Plug-in control, you hide the control. By embedding a Plug-in in your page that has its size set to width=2 height=0, no image appears on your Web page. You can control the Plug-in with JavaScript.

For example:

```
<script Language=JavaScript>
function playSource()
{if (navigator.appName == "Netscape")
  {document.javaPlug1.DoPlayPause();}
else
  {RAOCX.DoPlayPause();}
}
</script>
<A HREF="#" onClick="playSource()"><IMG
  RC="button.gif"></A>
<OBJECT ID=RAOCX CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-
  0020AFBBCCFA" WIDTH=2 HEIGHT=0>
<PARAM NAME="SRC" VALUE="pn://audio.real.com/welcome.rm">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
<embed src="start.rpm" Width=2 Height=0
  Controls=PlayButton name=javaPlug1>
</OBJECT>
```

The above example works in both Netscape and Internet Explorer. Only one control will appear on the Web page.

Making Controls Work Together

You can embed any number of RealPlayer controls within a Web page. Normally, each tag instance refers to different content. Sometimes, however,

you may want to link two or more controls together. For example, you can create a play button and an image window that work as a pair.

To include multiple components that work together, specify a **CONSOLE** attribute for each control and assign this the same value in each instance. For example:

```
<EMBED SRC="sample1.rpm" WIDTH=30 HEIGHT=33  
  CONTROLS="PlayButton" CONSOLE="Clip1">  
<EMBED SRC="empty1.rpm" WIDTH=176 HEIGHT=144  
  CONTROLS="ImageWindow" CONSOLE="Clip1">
```

Note Each **<EMBED>** tag must have a unique **SRC** attribute (if the same source is specified for two tags, one is ignored). Create a dummy .rpm file (named, for example: empty1.rpm) for a second control accessing a clip. Put a hard carriage return (ASCII code 13) into the dummy file.

Specifying a **CONSOLE** value of “_master” links a particular control to all other RealPlayer Controls on the page. Use this value, for example, to add a Status Bar to display information for all clips, for example:

```
<EMBED SRC="sample1.rpm" WIDTH=300 HEIGHT=33  
  CONTROLS="StatusBar" CONSOLE="_master">
```

Because many platforms, including Windows, only support one volume setting, all volume sliders act on the same underlying value and affect all clips regardless of **CONSOLE** name. For this reason, you may want to include only one volume slider per page, with no reference to a .rm file in its associated .rpm file.

If you only include an ImageWindow control on the Web page, the video stream may still be controlled by using the context sensitive pop-up menu. For Windows users, right-click the ImageWindow; Macintosh users, click and hold the mouse button until the pop-up menu appears.

Synchronized Multimedia

In addition to basic audio and video content, you can create real-time on-demand multimedia presentations. These presentations can be as simple as a narrated slide show of your home page or as intricate as a multi-frame training program that the viewer controls.

You can synchronize World Wide Web pages with clips. Thus the clip can be used as a “time line” to display new pages or frames in the Web browser or to update its content. This enables the creation of Internet slide shows, presentations, guided tours and site walk-throughs. A user can have full random access (fast forward and rewind), and the Web browser content is synchronized with the audio.

The information for the synchronized events is stored in one file that RealServer uses to stream clips and event information to RealPlayer. The event information is streamed to RealPlayer, and in turn RealPlayer sends Web page information to the Web browser telling it to update the page’s content.

RealServer’s .rm file format supports image maps and synchronized event information in a single file. Image maps allows users to interact with video content using “click-able” image maps. For more information, refer to “Image Maps” on page 218. For more information about synchronized events, refer to “Creating a RealVideo Events File” on page 350, and “Creating a RealAudio Events File” on page 352. RMMerge permits you to merge multiple .rm files together as one, including image maps and event information. For more information about RMMerge, refer to “Editing RealVideo files” on page 210.

Another way to create a synchronized multi-media presentation is using RealPlayer Plug-in. However, since sending the Web browser to a new URL unloads RealPlayer Plug-in when the HTML page is unloaded, it is best to create separate frames for RealPlayer Controls and for the changing images. For more information about frames, refer to “Frames: The Basics” on page 355.

Creating a RealVideo Events File

Creating a synchronized multimedia event using .rm files use the RMMerge tool. This is a three-step process:

1. Use a text editor to create an input file specifying the display time for each URL, title, author or copyright event.
2. Use the RMMerge tool to generate a binary file from the text input file by typing the following command:

```
rmmerge -f rmevents.dll event.txt output.rm
```

where:

event is the text file created in step 1
output is the .rm file that contains these events

3. Merge your new events .rm file with an audio or video file using the following command:

```
rmmerge <event file> <audio or video file> <output file>
```

where:

event file is the .rm file created in the previous step
audio/video is the file merged with the event file
output file is the file resulting from the merge

For example:

```
rmmerge event.rm audio_video.rm output.rm
```

Creating the Input File

Begin by creating a list of the URLs, titles, authors, or copyrights that you want to be shown during your presentation and the times within RealAudio or RealVideo clip when they should be displayed. The syntax for each entry should follow the format (with a space between each part of the command):

```
u starttime endtime EventURL  
i starttime endtime Title
```

```
a starttime endtime Author
c starttime endtime Copyright
```

where:

- u stands for URL event; each line starts with the letter u
- i stands for title; each line starts with the letter i
- a stands for author; each line starts with the letter a
- c stands for copyright; each line starts with the letter c
- starttime is the time into the clip when the new event is shown
- endtime is the time into the clip when that event ends
- EventURL (generally beginning with "http:" or "file:") is the Internet address for that event (usually an HTML document)

The time for starttime and endtime is:

```
[ [ [days:]hours:]minutes:]seconds[.tenths]
```

The lines of the input file must be in ascending order of start time. The end time should be at least one tenth of a second before the start time of the next event. The following example shows how an input file might look:

```
u 00:00:10.0 00:00:59.9 http://www.real.com/
u 00:01:00.0 00:02:00.0 http://www.mysite.com/ page2/
```

This input file tells RealPlayer to send the Web browser to the Progressive Networks home page ten seconds into the audio clip. One minute into the audio clip, the Web browser displays a page from "www.mysite.com".

The input file may also contain comment lines beginning with the # symbol. These comment lines are ignored by the event creation tool and are a good way to document the date that the file was created and the type of information found on each page.

Creating a RealAudio Events File

The RealAudio system allows you to create real-time on-demand multimedia presentations using the Cevents utility (Windows or UNIX). Creating a synchronized multimedia event using the Cevents command line tool is a three-step process:

1. Use a text editor to create an input file specifying the display time for each URL.
2. Use the Cevents command line tool to generate a binary file from the text input file.
3. Place the resulting .rae file in the same directory as the .ra audio file.

Creating the Input File

Begin by creating a list of the URLs that you want to be shown during your presentation and the times within the RealAudio clip when they should be displayed. The syntax for each entry should follow the format (with a space between each part of the command):

```
u starttime endtime EventURL
```

where:

u stands for URL event; each line starts with the letter u
starttime is the time into the audio clip when the new event is shown
endtime is the time into the audio clip when that event ends
EventURL (generally beginning with “http:” or “file:”) is the Internet address for that event (usually an HTML document)

The time for starttime and endtime is:

```
[ [days:]hours:]minutes:]seconds[.tenths]
```

The lines of the input file must be in ascending order of start time. The end time should be at least one tenth of a second before the start time of the next event. The following example shows how an input file might look:


```
u 00:00:10.0 00:00:59.9 http://www.real.com/  
u 00:01:00.0 00:02:00.0 http://www.mysite.com/  
page2/
```

This input file tells RealPlayer to send the Web browser to the Progressive Networks home page ten seconds into the clip. One minute into the clip, the Web browser displays a page from "www.mysite.com".

The input file may also contain comment lines beginning with the # symbol. These comment lines are ignored by the event creation tool and are a good way to document the date that the file was created and the type of information found on each page.

Generating an Event File

After creating the text version of the input file, you must convert the file to a binary event file. This is done with the command line utility Cevents. It uses the following syntax:

```
cevents <input text file> <output event file>
```

where:

<input event file> is the file you just created

<output event file> is the same name as the file with which it will be associated.

For example, to create synchronized multimedia presentation to accompany paradise.ra, you would generate paradise.rae using the following command:

```
cevents paradise.txt paradise.rae
```

Placing Event Files on Your Server

Copy the .rae file to the same directory as your .ra file. The .ra and .rae files must have the same name except for the file extension. RealServer automatically detects the file and send the event information to RealPlayer, which then sends it the Web browser.

Using Synchronized Multimedia from Local Files

RealPlayer can also read local .rae files just as the Server does. In order for the local presentation to work with multiple platforms and with both Internet Explorer and Netscape Navigator, without hardcoding the directory structure, you need to follow these steps.

To play a .rae file locally:

1. Place all .rpm/.ram, .ra, HTML and image files in one directory. The starting document should be named Index.html.
2. Do not use /'s in your file names. Use the following syntax:

```
u 10.0 45.0 &&media&&test2.html
```

Note You cannot use relative path commands like ../ if you want the presentation work on a Macintosh.

.TXT file - convert into .rae file using Cevents compiler

```
u 10.0 45.0 &&media&&test2.html
```

.RAM and .RPM

```
file:lshapfut.ra
```

IMAGES

```
<IMG src="lshapfut.gif">
```

HREFs

```
<a href="rasld011.htm">
```

EMBEDs

```
<EMBED SRC="lecture5.rpm">
```

Using Synchronized Multimedia with Bandwidth Negotiation

When you deliver a Synchronized Multimedia presentation using Bandwidth Negotiation, you can choose to deliver the same event for all encoding formats, or you can choose to deliver different events for each encoding format.

You can create a single synchronized multimedia presentation for multiple formats, or you can create a different multimedia presentation for each encoding format. If you are working with previously created content, you can continue to use the previous method to create presentations. For new content, refer to “Synchronized Multimedia” on page 349.

For general information about Bandwidth Negotiation, refer to “Bandwidth Negotiation” on page 223.

Frames: The Basics

Some Web browsers support a feature which allows the browser window to be divided into multiple regions. Each region is called a frame and can display a separate URL such as a graphic or document. For an introduction to frames visit Netscape’s Web site at:

<http://www.netscape.com>

If you are unfamiliar with frames the following two sites provide a good introduction:

Understanding Frames

http://www.netscape.com/navigate/understanding_frames.html

Frames: An Introduction

http://www.netscape.com/assist/net_sites/frames.html

Frame Document

A frame document describes the sub-HTML documents or frames that make up a window. The basic structure of a frame document is similar to that of a normal HTML document except that the FRAMESET tag replaces the BODY tag. Each frame is defined by the FRAME tag. In order to effectively use RealVideo, each FRAME tag needs the SRC and NAME attributes. The SRC points to the URL to be displayed in the frame. The NAME attribute assigns a name to the frame so that it can be targeted by links in other documents. The example below shows a simple frame document that would create two frames:

```
<HTML>
<FRAMESET ROWS="105,*">
  <FRAME SRC="banner.html" NAME="banner">
  <FRAME SRC="lyrics.html" NAME="Lyric">
</FRAMESET>
</HTML>
```

Synchronizing Frames and Video and Audio

Frames and RealVideo and RealAudio content are synchronized in the same way that a regular Web page and RealVideo and RealAudio content are synchronized: a .rae file. The difference lies in the addition of the targeted frame name to the text file that is used to create the .rae file. The syntax for each entry should follow the format:

```
u starttime endtime &&framename&&EventURL
```

where:

- u stands for URL event (each line starts with the letter u)
- starttime is the time into the clip when the new event is shown
- endtime is the time into the clip when that event ends, && is a delimiter
- framename is the frame name as specified in your frame document
- EventURL (generally beginning with "http:" or "file:") is the URL for that event (usually an HTML document).

The time format is:

```
[ [ [days:]hours:]minutes:]seconds[.tenths]
```

The lines of the input file must be in ascending order of start time. The end time should be at least one tenth of a second before the start time for the next event.

The example below shows an .rae input file for a Web site that displays lyrics as a song plays. Each verse is displayed within a frame called Lyric.

```
u 00:00:10.0 00:00:35.0
&&Lyric&&http://www.songs.com/ver1.html
u 00:00:35.0 00:00:50.0
&&Lyric&&http://www.songs.com/ver2.html
```

Note You cannot use frames within a presentation using RealPlayer. You can only use frames within a multimedia presentation using the Plug-in or ActiveX.

More Information on Frames

A complete discussion of frames is beyond the scope of this manual. For more information on frame attributes and syntax and on targeting windows consult the following Web pages:

Frames -- Syntax

http://www.netscape.com/assist/net_sites/frame_syntax.html

Targeting Windows

<http://www.netscape.com/eng/mozilla/2.0/relnotes/demo/target.html>

Index

A

Access log
 changing, 136
 contents, 138
 location, 93
 reading, 136
Access Logging, configuring, 91
Account
 creating on RealServers, 130
ActiveX
 advanced control attributes, 346
 making controls work together, 347
 methods and properties, 342
 RealVideo Control, 336
Administration
 messages sent through e-mail, 100
Adobe Premiere, 188
 encoding, 189
 marker options, 190
Advanced Control Attributes, 346
Advanced Server Settings, configuring, 115
AppleScripts, 254
Archiving live broadcasts, 304
AREA tag, 219
Audio
 algorithms, 199
 quality, 183
Author. *See* File Descriptions
AVI files, 180

B

Bandwidth
 threshold notification, 100
Bandwidth negotiation, 185, 223, 232
 creating file structure, 276
 delivering content, 223, 271
 diagram, 223, 272

 example, 226, 274, 276
 explained, 271
 raconv utility, 168, 276
 using during live broadcasts, 282
 utility program, 276
 with Synchronized Multimedia, 355
BandwidthEncoding setting, 309
BasePath setting, 79
Batch Encoding, 261
Broadcast. *See* Live broadcast
Broadcasting, live, 21

C

Cevents, 353
 command reference, 165
ClientConnections setting, 103
Clients window, System Manager, 161
Clips starting automatically, 347
Command line Encoding
 Windows, 250
Command reference
 cevents, 164, 165
 kill, 166
 pnserver, 167
 raconv, 164, 168
 rafile, 164, 169, 171
 rasm, 173
 slta, 164, 177
Configuration File, 67
 basic settings, 77
 editing, 70
 editing with System Manager, 71
 general settings, 78
 UNIX sample, 69
 Windows NT sample, 68
Configuration settings
 BandwidthEncoding, 309
 BasePath, 79
 ClientConnections, 103

- ConnectControlList, 105, 134
- CustomerName, 80
- DefaultErrorFile, 81
- EncoderPassword, 282
- EncoderTimeout, 283
- ErrorLogPath, 92
- Group, 81
- InputFile, 312
- LicenseClients, 127
- LicenseKey, 82
- LiveFileBandwidthNegotiation, 310
- LiveFilePassword, 306
- LiveFileSize, 306
- LiveFileTarget, 307
- LiveFileTime, 308
- Local host, 83
- LoggingStyle, 93
- LogPath, 92
- MailMessageLevel, 96
- MailMessageLimit, 97
- MailMessageSMTPHost, 98
- MailMessageUser, 98
- MailUsageCC, 99
- MailUsagePeriod, 100
- MailUsageThreshold, 101
- MaxBandwidth, 104
- MaxThreads, 84
- MinPlayerProtocol, 84
- MonitorConnections, 107
- MonitorPassword, 108
- MulticastAddressRange, 320
- MulticastControlList, 321
- MulticastDeliveryOnly, 323
- MulticastPort, 323
- MulticastTTL, 324
- OutputFile, 312
- PidPath, 86
- PnaPort, 85
- RemoteLicenseHost, 128
- RemoteLicensePort, 129
- ResolverPort, 88
- ServerHost, 313
- ServerPassword, 313
- ServerPort, 314
- SplitterBufferDelay, 122
- SplitterControlList, 121

- StatsMask, 94
- Timeout, 86
- URL, 308
- User, 88
- UserDir, 132
- UserList, 133
- Configuration settings, changing, 70, 71
- Configuring Access and Error Logging, 91
- Configuring Advanced Server Settings, 115
- Configuring E-mail and Usage Thresholds, 95
- Configuring Web Servers to Work with RealServer, 109
- Configuring Your Web Site, 326
- ConnectControlList setting, 105, 134
- Connections
 - threshold notification, 100
- Control Attributes, advanced, 346
- Controls
 - autostart, 347
 - display, 338
 - no text labels, 346
- Conventions used in manual, 12
- Copying RealAudio Files
 - Macintosh, 265
 - UNIX, 263
 - Windows, 263
- Copying RealVideo Files
 - UNIX, 212
 - Windows, 212
- Copyright. *See* File Descriptions
- Custom Controls for RealVideo, 332
- CustomerName setting, 80
- Customizing Calls to Audio Content, 329
- Customizing Calls to Video and Audio Content, 329

D

- DefaultErrorFile setting, 81
- Delivering a RealVideo Clip, 17
- Delivering Live RealVideo Content, 284
- Delivering Synchronized Multimedia, 18
- Disk space required, 23
- Drag-and-Drop Encoding
 - Macintosh Encoder, 254
 - Windows Encoder, 195, 249

DURATION tag, 218

E

- Editing Input Files, 202, 262
- Editing RealAudio Files, 262
- Editing RealVideo Files
 - UNIX, 210
 - Windows, 202
- Editing utilities, 15
- E-mail
 - notification messages sent, 100
- e-mail address, configuring, 95
- Embedded Object Parameters, 336
- EncoderPassword setting, 282
- EncoderTimeout setting, 283
- Encoding a RealVideo clip, 179
- Encoding a static file
 - Macintosh Encoder, 251
 - Windows Encoder, 245
- Encoding Algorithms, 232
- Encoding live content
 - Macintosh Live Encoder, 298
 - UNIX Live Encoder, 300
- Encoding RealAudio files
 - batch, 261
 - RealAudio Encoder for Macintosh, 250
 - RealAudio Encoder for UNIX, 258
 - RealAudio Encoder for Windows, 244
 - Sound Forge, 234
- Encoding RealAudio live content
 - Windows Live Encoder, 296
- Encoding RealVideo files
 - Adobe Premiere, 188
 - Windows, 191
- Error log
 - changing, 136
 - default, 81
 - described, 144
 - reading, 136
- Error Logging, configuring, 91
- Error LogPath setting, 92
- Error message
 - document not found, 146
 - Out of Memory, 146
 - SIGPIPE signal received, 146

Events File, creating, 350, 355

F

- Filename Extensions, 329
- Files
 - .ra, 16
 - .ram, 16, 222
 - .rm, 16
 - .rpm, 16
 - delivering different files using Bandwidth Negotiation, 271
 - delivering with Bandwidth Negotiation, 223
 - descriptions, modifying, 222
 - formats, 225, 226
 - log, 136, 138
 - RealServer, 223, 271
- Files window, System Manager, 162
- Formats, video, 182
- Frames, 355
 - frame document, 356
 - synchronizing with audio, 356

G

- Graph window, System Manager, 162
- Group setting, 81

H

- Hidden Controls, 347
- Hosting
 - server, 129
- HTTP Port 80, 89
- HTTP Streaming, 331

I

- Image
 - cropping, 194, 291
 - quality, 182
 - size, 182
- Image Map, 218
 - merging with video image, 221
- Input File, creating, 350

InputFile setting, 312

Installing

- UNIX Server from CD-ROM, 50, 51

- Windows Encoder, 46

- Windows Live Encoder, 46

Internet service providers

- creating accounts, 130

Intranet

- guaranteeing streams to users, 130

Invisible Controls, 347

J

Java and JavaScript

- callback methods, 345

- methods, 344

K

Kill command reference, 166

L

License key

- configuring a license client, 126

- configuring a license server, 126

- sharing among multiple computers, 125

LicenseClients setting, 127

LicenseKey setting, 82

Limiting Server Bandwidth and the Number

- of Simultaneous Connections, 102

Live

- broadcasting, 21

- capture-to-file, 183

- encoding, 21

- system requirements, 26

Live broadcast

- archiving live events, 304

- connecting to RealAudio Encoder, 281

- example, 278

- splitting a live media stream, 116

Live RealAudio Encoder

- UNIX, 300

Live RealAudio Encoder, Windows, 296

Live RealAudio Encoder, Macintosh, 298

Live RealVideo Content, 284

- command line controls, 293

- high quality, 286

- software requirements, 286

- system requirements, 284

- templates, 288

Live RealVideo Encoder

- Windows, 288

LiveFileBandwidthNegotiation setting, 310

LiveFilePassword setting, 306

LiveFileSize setting, 306

LiveFileTarget setting, 307

LiveFileTime setting, 308

LocalHost setting, 83

Log files

- changing, 136

- format, 138

- reading, 136

LoggingStyle setting, 93

LogPath setting, 92

M

Macintosh

- copying files, 265

- pasting files, 267

- RACut, 265

- RAPaste, 267

Macintosh Encoder, 250

- AppleScripts, 254

- drag-and-drop encoding, 254

- encoding a static file, 251

- Preferences window, 252

MailMessageLevel setting, 96

MailMessageLimit setting, 97

MailMessageSMTPHost, 98

MailMessageUser setting, 98

MailUsageCC setting, 99

MailUsagePeriod setting, 100

MailUsageThreshold setting, 101

Managing Content

- RealServer, 271

Managing Content, 223

MAP tag, 219

- /MAP tag, 221

MaxBandwidth setting, 104

MaxThreads setting, 84

- Memory
 - Out of Memory Message, 146
 - performance, 22
 - Metafiles, 16, 327
 - creating, 328
 - table of, 224
 - MIME types, configuring for use with
 - Progressive Networks streaming media system, 109
 - MinPlayerProtocol setting, 84
 - Modifying RealAudio File Descriptions, 268
 - Modifying RealVideo File Descriptions, 222
 - Monitor. *See also* System Manager
 - running on Windows, 160
 - MonitorConnections setting, 107
 - Monitoring performance
 - from Windows, 160
 - MonitorPassword setting, 108
 - MOV files, 182
 - MulticastAddressRange setting, 320
 - MulticastControlList setting, 321
 - MulticastDeliveryOnly setting, 323
 - Multicasting, 315
 - MulticastPort setting, 323
 - MulticastTTL setting, 324
 - Multiple computers
 - sharing license information, 125
- N**
- New connections
 - preventing, 158
- O**
- OutputFile setting, 312
- P**
- Packets
 - early, 141
 - late, 142
 - missing, 141
 - out-of-order, 141
 - total, 141
 - Pasting RealAudio Files
 - Macintosh, 267
 - UNIX, 264
 - Windows, 264
 - Performance
 - memory, 22
 - monitoring from Windows, 160
 - PidPath setting, 86
 - Plug-in
 - Adobe Premiere, 188
 - RealVideo, 16, 332
 - PnaPort setting, 85
 - Pnserver
 - command reference, 167
 - Preferences Window
 - Macintosh Encoder, 252
 - Process ID
 - changing the Server configuration, 166
 - log, 86
 - stopping the server, 159
 - Producing High Quality Video, 182
 - Producing Higher Quality Sound, 231
 - ps command, 166
- Q**
- QuickTime files, 182
- R**
- ra file, 16
 - Raconv command reference, 164, 168
 - Racut, 263, 265
 - Rafile
 - command reference, 169, 171
 - using, 304
 - ram file, 16, 222
 - Rapaste, 264, 267
 - Rasm. *See also* System Manager
 - command reference, 173
 - Rax Tool, 269
 - Real Icon, 11, 326
 - Real Server
 - Log files, 136
 - RealAudio clip, 16
 - RealAudio Encoder
 - connecting to RealServer, 281

- Macintosh, 250
 - UNIX, 258
 - Windows, 244
 - RealAudio Live Encoder
 - Macintosh, 298
 - Macintosh, encoding live content, 298
 - UNIX, 300
 - Windows, 296
 - Windows, encoding live content, 296
 - RealAudio Player
 - using to test RealAudio Server, 163
 - RealAudio Server
 - starting automatically on UNIX, 157
 - starting automatically on Windows NT, 156
 - starting manually on UNIX, 156
 - stopping gracefully, 158
 - RealMedia clip, 16
 - RealPlayer, 15
 - using to test RealServer, 34, 56
 - RealServer, 15
 - connection to System Manager, 71
 - installing on Windows NT from Internet, 41
 - managing content, 223, 271
 - setting up for live broadcasts, 281
 - sharing License key among multiple computers, 125
 - stopping, 34
 - testing, 34, 56
 - RealServer hosting, 129
 - RealVideo clip, 14
 - encoding, 179
 - updating, 21
 - RealVideo Control for ActiveX, 336
 - RealVideo Encoder, 15
 - Windows, 191
 - RealVideo Live
 - software requirements, 286
 - system requirements, 284
 - RealVideo Live Encoder
 - Windows, 288
 - RealVideo Plug-in, 16, 332
 - RealVideo Tools, 15
 - Recording live broadcasts, 304
 - RemoteLicenseHost setting, 128
 - RemoteLicensePort setting, 129
 - Resetting peak usage, using System Manager, 162
 - ResolverPort setting, 88
 - rm files, 16
 - RMCut, 212
 - RMDump, 213
 - RMEdit, 210, 222
 - RMMerge, 221, 350
 - RMTools, 222
 - editing file data, 206
 - editing file properties, 203
 - examining file contents, 205
 - rpm file, 16
- ## S
- sample configuration file
 - UNIX, 69
 - Windows NT, 68
 - Sampling Rates
 - valid, 230
 - Saving live broadcasts, 304
 - Security, 104
 - Server Bandwidth, limiting, 102
 - Server Requirements
 - CPU usage, 22
 - disk space, 23
 - memory, 22
 - Server Settings, configuring advanced, 115
 - Server upgrade
 - steps before installing new version, 39, 61
 - Server.cfg, 79
 - ServerHost setting, 313
 - ServerPassword setting, 313
 - ServerPort setting, 314
 - Settings File
 - example, 197
 - Shutting down
 - gracefully, 158
 - SIGPIPE signal, 146
 - Simulated Live Broadcast, 311
 - Simultaneous Connections, limiting, 102
 - Slt, 311
 - command reference, 177
 - using, 311
 - Smart Networking, 89

- Sound Cards, 231
 - Sound Forge, 234
 - Sound Quality, 231
 - Source Files, 179
 - types, 230
 - SplitterBufferDelay setting, 122
 - SplitterControlList setting, 121
 - Splitting, 115
 - configuring, 116
 - Splitting a live media stream, 116
 - Controlling splitter access to a server, 119
 - Starting RealAudio Server
 - automatically on UNIX, 157
 - automatically on Windows NT, 156
 - manually on UNIX, 156
 - Starting RealServer
 - manually on Windows NT, 33, 55
 - StatsMask setting, 94
 - Stopping
 - gracefully, 158
 - RealAudio Server, 158
 - RealServer, 34
 - Streams
 - threshold notification, 100
 - Synchronized Multimedia, 349
 - Cevents, 353
 - frames, 355
 - from local files, 354
 - placing on a Server, 353
 - using with Bandwidth Negotiation, 355
 - Synchronized Multimedia, delivering, 18
 - System and Software requirements. *See* Server Requirements
 - Windows Encoder, 25, 28
 - Windows Live Encoder, 25, 28
 - System Manager
 - clients window, 161
 - connecting to RealServer, 71
 - editing configuration file, 71
 - files window, 162
 - graph window, 162
 - maximum connections, 107
 - monitoring performance, 160
 - resetting peak usage, 162
 - System Manager Settings, 107
 - System requirements
 - live encoding, 26
- ## T
- Templates, 185, 198
 - create new, 200
 - live RealVideo content, 288
 - pre-defined, 186
 - Testing RealServer, 34, 56
 - threads, maximum, 84
 - Threshold notification e-mail, 99
 - Timeout setting, 86
 - Title. *See* File Descriptions
 - Troubleshooting, 151
- ## U
- uncompressing .tar files, 49
 - UNIX
 - boot-time scripts, 157
 - copying files, 212, 263
 - editing files, 210
 - kill, 166
 - pasting files, 264
 - ps command, 166
 - Racut, 263
 - Rapaste, 264
 - RMCut, 212
 - RMDump, 213
 - RMEdit, 210
 - tar files, 49
 - UNIX Encoder, 258
 - encoding a static file, 258
 - UNIX Live Encoder
 - encoding live content, 300
 - UNIX Server
 - changing configuration settings, 70, 71
 - changing log files, 137
 - installing RealServer from CD-ROM, 50, 51
 - starting RealAudio Server automatically, 157
 - stopping RealAudio Server, 159
 - Upgrading Server
 - steps before installing new version, 39, 61
 - URL setting, 308

- Usage Thresholds, configuring, 95
- User setting, 88
- UserDir setting, 132
- UserList setting, 133

V

- Video Formats, 182
- Video Quality, 182

W

- Web Browser, 15
- Web Server, 15
- Web servers
 - configuring for use with RealServer, 109
 - supported, 23
- Web Servers, configuring, 109
- Web Site Configuration, 326
- Windows
 - copying files, 212, 263
 - drag-and-drop encoding, 195
 - editing files, 210
 - editing RealVideo files, 202
 - encoding a static file, 192
 - pasting files, 264

- Racut, 263
- Rapaste, 264
- RealVideo Encoder, 191
- RMCut, 212
- RMEdit, 210

- Windows Encoder
 - command line encoding, 250
 - drag-and-drop encoding, 249
 - encoding a static file, 245
 - installing, 46
 - setting preferences, 247
 - system and software requirements, 25, 28
- Windows Encoding
 - command line encoder example settings file, 197
- Windows Live Encoder
 - installing, 46
 - system and software requirements, 25, 28
- Windows NT Server
 - changing log files, 137
 - installing RealServer from Internet, 41
 - starting RealAudio Server automatically, 156
 - starting RealServer manually, 33, 55
 - stopping RealAudio server, 159